



Universidad
Carlos III de Madrid

Departamento de Ingeniería de Sistemas y Automática

TRABAJO FIN DE GRADO

DISEÑO Y CONSTRUCCIÓN DE UN MICRO- ROBOT USANDO TARJETAS ARDUINO Y UNA IMPRESORA 3D

Autor: Ángel Mate Martínez

Tutor: Dr. Arturo de la Escalera Hueso



Agradecimientos

Primeramente agradecer a mi tutor, el Dr. Arturo de la Escalera por permitirme realizar este Trabajo Fin de Grado, por su ayuda, sus consejos y tiempo dedicado. También agradecer a Ángela Nombela Piqueras por su ayuda.

También quiero agradecer a mi familia por el apoyo que me llevan dando todos estos años a pesar de la distancia que nos separa, sin ellos no podría haber acabado la carrera. Y a mis amigos que en los momentos de mayor estrés siempre estaban ahí para darme su apoyo.



Resumen

El principal objetivo de este trabajo de fin de grado es la construcción de varios micro-robots como plataforma para la docencia.

Este trabajo constara de dos partes una mecánica y otra electrónica. En cuanto a la mecánica diseñaremos el robot basándonos en el skybot. Para fabricar las piezas diseñadas se empleara una impresora 3D.

En cuanto a la electrónica se emplearan varios tipos de sensores para aportar al robot una mayor flexibilidad. Se diseñara una shield impresa para la placa arduino con el acondicionamiento de dichos sensores y motores, ya que se utilizara arduino para la programación. En concreto se usara la tarjeta Arduino mega 2560.

Palabras clave: Micro-robot, PCB, Arduino, Impresión 3D, diseño, mecánico, electrónica, sensores.



Abstract

The main objective of this work is the construction of several micro-robots as a platform for teaching.

This work will consist of two parts mechanics and other electronics. As regards the mechanical design the robot based on the skybot. To make the designed pieces using a printer 3D.

As to the electronics, use various types of sensors to provide greater flexibility to the robot. They design a printed shield for arduino with conditioning of the sensors and motors, and to be used for programming arduino. Specifically, the board will use arduino mega 2560.

Keywords: *Micro-robot, PCB, arduino, printing 3D, design, mechanical, electronics, sensors.*



Índice

Capítulo 1: Fases y objetivos	10
1.1 Objetivo	11
1.2 Fases del desarrollo	11
Capítulo 2: Impresión 3D	14
2.1 Introducción	15
2.2 Tecnologías de impresión 3D	15
2.3 Tipos de impresoras 3D	18
Capítulo 3: Arduino	21
3.1 Introducción	22
3.2 Arduino mega 2560	22
3.3 Funciones básicas para programación	24
Capítulo 4: Sistema sensorial	26
4.1 Sensor de contacto o Bumpers	27
4.2 Fotorresistencia	28
4.3 Sensor CNY70	30
4.4 Sensor Sharp GP2Y0A2	31
4.5 Sensor de ultrasonidos.....	33
Capítulo 5: Software.....	37
5.1 OpenScad	38
5.2 Arduino software	39
5.3 Eagle	40
5.4 Slic3r y Pronterface.....	41
Capítulo 6: Sistema locomotor	43
6.1 Elección de los motores	44
6.2 Control de los motores	45
Capítulo 7: Diseño Electrónico.....	47
7.1 Introducción	48
7.2 Acondicionamiento de los sensores	49
7.2.1 Sensor de contacto o Bumper	49



7.2.2 Fotorresistencia o LDR	49
7.2.3 Sensor CNY70.....	50
7.2.4 Sensor Sharp	50
7.2.5 Sensor de Ultrasonidos	51
7.3 Diseño del circuito impreso.....	51
Capítulo 8: Diseño Mecánico	54
8.1 Introducción	55
8.2 Skybot	56
8.3 Diseño del robot	56
8.3.1 Base sensores.....	56
8.3.2 Base para el sensor Sharp y base del sensor de ultrasonidos.....	57
8.3.3 Base del sensor CNY70	57
8.3.4 Ruedas	57
8.3.5 Chasis del robot	58
8.4 Proceso de fabricación.....	59
Capítulo 9: Programación	62
9.1 Introducción	63
9.2 Calibración de sensores	63
9.3 Función de los sensores según la programación.....	64
9.3.1 Sensor de contacto y Fotorresistencia (LDR).....	64
9.3.2 Sensor CNY70 y sensor Sharp	64
9.3.3 Sensor de ultrasonidos.....	64
9.4 Posibles tareas que puede realizar el robot.....	65
Capítulo 10: Planificación y presupuesto	67
10.1 Planificación	68
10.1.1 Resumen de horas dedicadas y diagrama de Gantt	68
10.2 Costes de ejecución	70
10.2.1 Costes por material de laboratorio	70
10.2.2 Costes por software	70
10.2.3 Costes por hardware	71
10.2.4 Costes por personal	71



10.2.5 Coste total de presupuesto de ejecución.....	71
10.3 Importe total del presupuesto	72
Conclusiones y Trabajos Futuros	73
Capítulo 11: Bibliografías y Referencias	74
11.1 Referencias relacionadas con sensores y motores.....	75
11.2 Referencias relacionadas con la impresión 3D	75
11.3 Referencias relacionadas con el software	76
Capítulo 12: Conclusiones y Anexos	77
12.1 Hoja de características del sensor CNY70	78
12.2 Hoja de características del sensor Sharp	83
12.3 Hoja de características del driver L293	88
12.4 Esquemático de la shield	91

Índice de Figuras

Figura 1: Imagen del primer prototipo del micro-robot	13
Figura 2: Imagen de la flota de antiguos micro-robots.....	13
Figura 3: Imagen de una impresora 3D basada en la tecnología LOM	15
Figura 4: Esquema del principio de funcionamiento de la tecnología FDM	16
Figura 5: Esquema del principio de funcionamiento de la estereolitografía	17
Figura 6: Esquema del principio de funcionamiento de la tecnología SLS	18
Figura 7: Imagen de la impresora 3D de hielo	18
Figura 8: Imagen de la impresora 3D Thing-o-matic	19
Figura 9: Imagen de la impresora 3D RepRap Air	20
Figura 10: Imagen del logotipo Arduino	22
Figura 11: Imagen de la placa Arduino Mega 2560	23
Figura 12: Imagen de una batería para Arduino	24
Figura 13: Imagen del funcionamiento de un sensor tipo pulsador	27
Figura 14: Imagen de un sensor bumper	27
Figura 15: Imagen de una fotorresistencia	28
Figura 16: Grafica de LDR representando la resistencia frente a la Iluminación	29
Figura 17: Imagen de un sensor CNY70	30
Figura 18: Grafica de sensor CNY70 representando la intensidad frente a la distancia	31
Figura 19: Imagen de un sensor Sharp GP2Y0A2	31
Figura 20: Imagen de un conector JST	32
Figura 21: Grafica del sensor Sharp GP2Y0A2	32
Figura 22: Imagen de un sensor SRF05	33
Figura 23: Imagen de un sensor SRF05 en modo 1	33
Figura 24: Diagrama de tiempos del SRF05 modo 1.....	34
Figura 25: Imagen de un sensor SRF05 en modo 2	35
Figura 26: Diagrama de tiempos del SRF05 modo 2.....	35
Figura 27: Diagrama del haz del SRF05	36
Figura 28: Imagen de la interface OpenScad	38
Figura 29: Imagen de la interface del arduino software	39

Figura 30: Imagen de la interface de Eagle del Layout Editor	40
Figura 31: Imagen de la interface de Skeinforge	41
Figura 32: Imagen de la interface de Pronterface	42
Figura 33: Imagen del servo Futaba 3003 trucado	45
Figura 34: Imagen de un integrado L293DNE	45
Figura 35: Imagen del patillaje de un integrado L293DNE	46
Figura 36: Imagen conexionado de los motores al L293D y a placa Arduino	46
Figura 37: Imagen de las tarjetas Skypic (Izquierda) y Sky293 (derecha)	48
Figura 38: Imagen conexionado de un Bumper	49
Figura 39: Imagen conexionado de las LDR	49
Figura 40: Imagen conexionado de un CNY70	50
Figura 41: Imagen conexionado del sensor sharp	51
Figura 42: Imagen conexionado del sensor ultrasonido	51
Figura 43: Imagen del diseño de la placa impresa	52
Figura 44: Imagen palca impresa con componentes	53
Figura 45: Imagen conector servo tipo JR	53
Figura 46: Imagen de un Skybot	55
Figura 47: Parte trasera de la base sensores	56
Figura 48: Parte delantera de la base sensores	56
Figura 49: Imagen de la Base para sensor Sharp (izquierda) y base del sensor de ultrasonidos (derecha)	57
Figura 50: Imagen de la Base del sensor CNY70	57
Figura 51: Imagen de la rueda del robot	58
Figura 52: Chasis del robot	58
Figura 53: Imagen de impresión doble	60
Figura 54: Imagen piezas del robot impresas	61



Índice de Tablas

Tabla 1: Valores de impresión 3D	60
Tabla 2: Tiempo de impresión de un robot	61
Tabla 3: Calibración del sensor Sharp	63
Tabla 4: Calibración del sensor de luz (LDR)	64
Tabla 5: Costes por material de laboratorio	70
Tabla 6: Costes por software	70
Tabla 7: Costes por hardware	71
Tabla 8: Costes por personal	71
Tabla 9: Coste total del presupuesto de ejecución	71
Tabla 10: Importe total del presupuesto	72



Capítulo 1

Fases y objetivos

1.1 Objetivo

1.2 Fases del desarrollo

1.1 OBJETIVO

El objetivo de este Trabajo Fin de Grado es la construcción de un micro-robot para docencia. Se partirá de una modificación ya existente del robot Skybot, modificarla tanto mecánicamente como electrónicamente. Se le añadirá algún sensor más, se empleará Arduino para la electrónica y la programación y se cambiará el diseño del Robot. Fabricaremos una flota de varios robots que se utilizarán en la universidad con carácter didáctico, para cursos de Micro-Robótica.

Se pueden señalar los siguientes objetivos parciales:

1. Mecánicamente se modifica el chasis incorporando un soporte para un rodamiento que actuara como rueda loca. Independiente del chasis se creará una base para situar los sensores y la placa Arduino con el circuito impreso. Con esto se consigue que solo habrá que modificar la base de los sensores si se quiere incorporar alguno más, manteniendo el chasis.
2. Electrónicamente se eliminan las tarjetas SKYPIC y SKY293 ya que se utilizará la placa Arduino y se diseñará una shield para la Arduino en la que se situará el acondicionamiento de los sensores y el control de los motores.
3. Se incorporará como principal novedad un sensor infrarrojo Sharp para detectar obstáculos y la distancia a la que se encuentran y dirigirse hacia ellos o parar para no chocar.

Con todo esto lo que se pretende es diseñar un robot con más funcionalidades, open source e imprimible con una impresora 3D desde cualquier parte del mundo.

1.2 FASES DEL DESARROLLO

Las fases que hemos seguido para realizar este proyecto han sido las siguientes:

1. Primera Fase:

En esta fase se buscará información sobre el skybot y la modificación realizada sobre la que se basará. Se decidirá los sensores que se van a usar y se establecerán los objetivos para planificar el proyecto y realizar este con éxito.

2. Segunda Fase:

Una vez que hemos recopilado la información y se ha establecido los objetivos, se empezara a estudiar a fondo como vamos a diseñar mecánicamente el robot y también se empezara a recopilar información sobre la placa Arduino.

3. Tercera Fase:

El siguiente paso es programar las piezas utilizando un programa de CAD. En este caso el software utilizado es el openscad. También en esta fase se estudiara que acondicionamiento es necesario para conectar los sensores a la placa Arduino.

4. Cuarta Fase:

Una vez que sabemos cómo hay que acondicionar los sensores se empezara a diseñar el esquemático de la shield (placa impresa) que vamos a crear. Para ello se utilizara el software EAGLE que previamente hemos estudiado como funciona.

5. Quinta Fase:

En este paso lo que se hará será empezar a programar con Arduino una vez que se han realizado las conexiones desde dos protoboard, previamente a esto se habrá estudiado como vamos a controlar los motores y que sensores necesitan calibración o no. Se continuara con la creación de la Shield ya que una vez que se ha realizado el esquemático pasamos a hacer el Board.

6. Sexta Fase:

Una vez que se tiene diseñada la shield, se crean los archivos Gerber para mandar hacer la shield a la empresa. También se comenzara con la impresión en 3D. Para ello se debe aprender a utilizar correctamente la impresora 3D ya que existen muchos parámetros para obtener distintos tipos de impresión. Decir que he participado en el montaje de un modelo de la impresora 3D que se va a utilizar.

7. Séptima Fase:

Una vez que se tiene impreso el primer robot, fabricada la shield con los componentes soldados, se comenzara a montar el primer robot para probarlo y si hay que hacer modificaciones hacerlas para posteriormente hacer el resto de los robots.

8. Octava Fase:

Una vez que se han corregido los fallos del primer prototipo se mandan hacer las 9 placas impresas para los robots futuros. Y se inicia la impresión 3D de las piezas para realizar más robots.

9. Novena Fase:

Se procede a soldar los componentes de las 9 placas impresas y al montaje de los robots finalizando así con las fases del desarrollo de este trabajo de fin de grado.

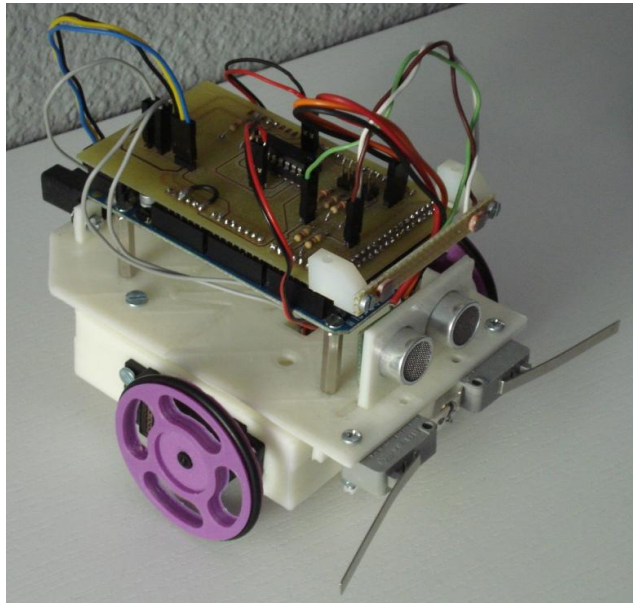


Figura 1: Imagen del primer prototipo del micro-robot [Fuente: Elaboración propia]

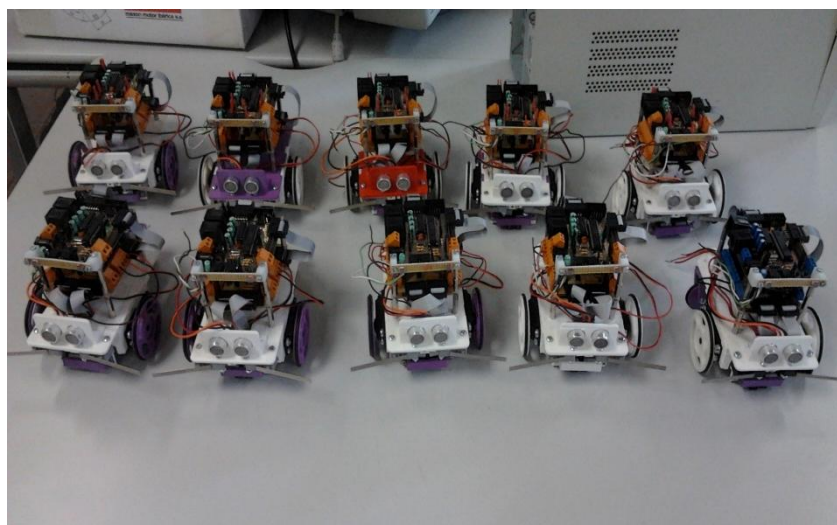


Figura 2: Imagen de la flota de antiguos micro-robots



Capítulo 2

Impresión 3D

2.1 Introducción

2.2 Tecnologías de impresión 3D

2.3 Tipos de impresoras 3D

2.1 INTRODUCCIÓN

La tecnología de impresión 3D se inicia en 1987 por la empresa norteamericana 3D Systems, que pone en marcha el sistema conocido como estereolitografía. Años más tarde comienzan a surgir nuevos sistemas de impresión 3D hasta llegar el momento actual en que existen impresoras 3D caseras como se verá más tarde. Desde el 2003 ha habido un incremento en las ventas de impresoras 3D, una de las causas de este incremento puede deberse a la disminución de los costes de las mismas.

La impresión 3D es un grupo de tecnologías de fabricación por adición donde una pieza u objeto tridimensional se crea mediante la superposición de capas sucesivas de material.

Una impresora 3D es una máquina capaz de realizar impresiones de diseños en 3D, creando piezas o maquetas volumétricas a partir de un diseño hecho por ordenador, es decir se convierte un archivo CAD en un prototipo real. Por ello cada vez se utiliza en mas sectores tales como en joyería, medicina, arquitectura, ingeniería etc.

2.2 TECNOLOGIAS DE IMPRESIÓN 3D

Principalmente lo que se diferencian entre unas y otras tecnologías es en la forma en la que se usan las diferentes capas y en el material.

Las principales tecnologías son:

Fabricación por corte y laminado (LOM):

Sistema de impresión 3D basado en la continua colocación de láminas de material sobre una plataforma, posteriormente se prensa cada una de ellas con un rodillo caliente que la adhiere a la hoja anterior.



Figura 3: Imagen de una impresora 3D basada en la tecnología LOM [Fuente: <http://spanish.alibaba.com>]

Modelado por deposición fundida (FDM):

Es una tecnología que fue desarrollada por Stratasys a finales de 1980, se emplea para la fabricación rápida de prototipos. Su funcionamiento se basa en un filamento de plástico o metal enrollado a una bobina este es fundido mediante una boquilla caliente que se puede mover en ambas direcciones, horizontal y vertical, mediante un mecanismo, está a su vez es controlada a través del software de un ordenador donde se tendrá nuestro diseño. Este filamento se solidifica instantáneamente sobre la base en la que se deposita.

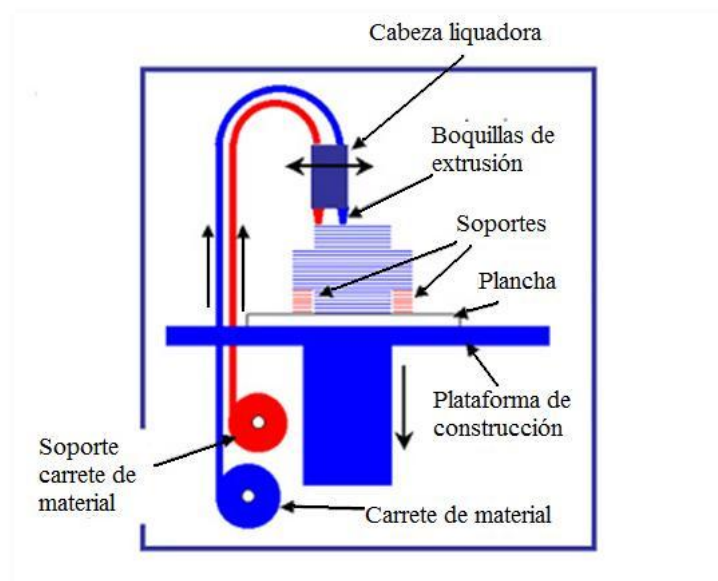


Figura 4: Esquema del principio de funcionamiento de la tecnología FDM [Fuente: <http://tfmrimuned.wordpress.com>]

De los diferentes polímeros que se pueden emplear quiero destacar el acrilonitrilo butadieno estireno (ABS) que es el que se emplea para la fabricación de este micro-robot.

Fotopolimerización:

Se utiliza principalmente en la estereolitografía (SLA) que consiste en proyectar un laser UV sobre un baño de resina fotosensible líquida para polimerizarla.

Una tecnología similar a la estereolitografía es la fotopolimerización por luz UV (SGC), funciona mediante la solidificación de un fotopolímero o resina fotosensible con una lámpara de UV de gran potencia.

El procesamiento digital por luz (DLP) trata de solidificar por la acción de la luz una resina en estado líquido por el sistema de proyección por máscara. El polímero líquido expuesto endurece, posteriormente la placa baja en pequeños incrementos y el polímero es expuesto de nuevo a la luz, así hasta que se finaliza el modelo.

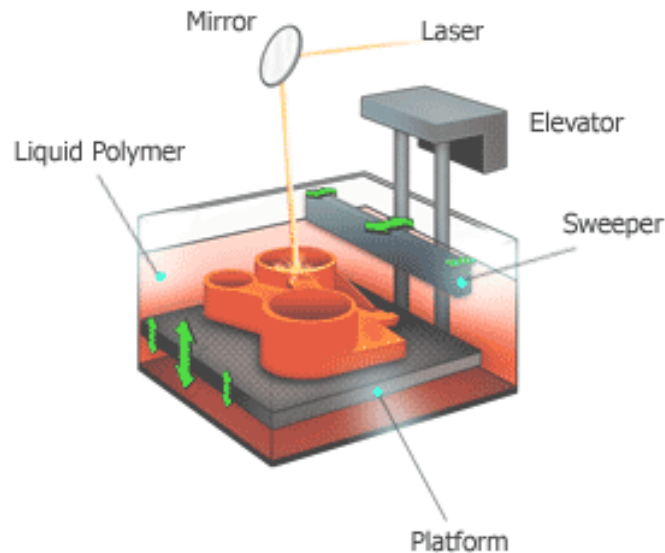


Figura 5: Esquema del principio de funcionamiento de la estereolitografía [Fuente: www.tallerdesoluciones.blogspot.com.ar]

Proyección aglutinante (DSPC):

Es una tecnología de impresión 3D que trabaja mediante la deposición de material en polvo (composite) en capas que se cohesiona mediante la impresión de un “chorro de tinta” de un material aglutinante.

Sinterización selectiva laser (SLS):

Se deposita una capa de polvo, de unas decimas de mm en una cuba que se ha calentado previamente a una temperatura un poco inferior al punto de fusión del polvo. Posteriormente un laser CO2 sinteriza el polvo en puntos concretos para la creación de la pieza.

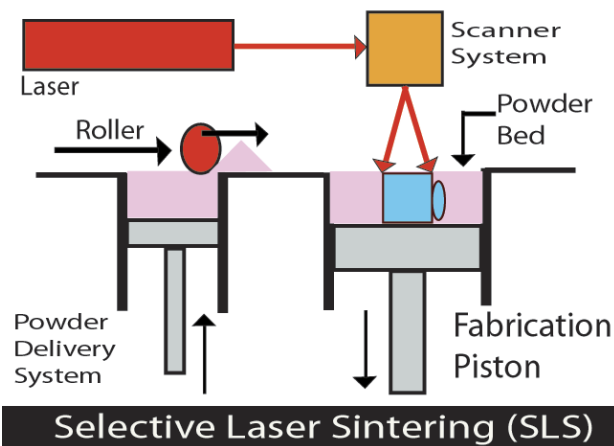


Figura 6: Esquema del principio de funcionamiento de la tecnología SLS [Fuente: www.martello.co.uk]

Impresión con hielo:

Esta impresión es bastante novedosa y consiste en que por medio de un enfriamiento controlado de agua tratada son capaces de fabricar una autentica impresión 3D con hielo. En la universidad de McGill de Montreal (Canadá) tienen un proyecto relacionado con esta tecnología, en la imagen siguiente podemos ver la impresora que han creado.

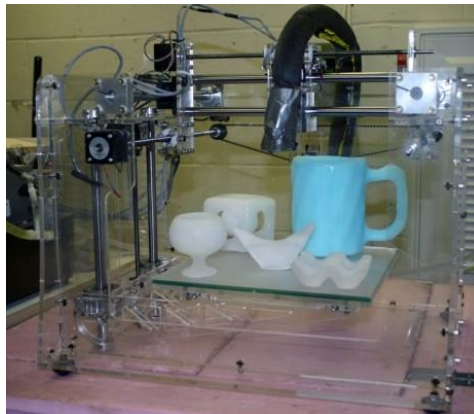


Figura 7: Imagen de la impresora 3D de hielo [Fuente: www.gizig.com]

2.3 TIPOS DE IMPRESORAS 3D

Ahora se exponen los 2 tipos de impresora que se ha empleado en este proyecto, una comprada a una empresa y otra casera.

Thing-o-matic:

Pertenece a la empresa Makerbot Industries, la cual fue fundada en enero de 2009 por Bre Pettis, Mayer Adán y Zach “Hoeken” Smith y tiene su sede en Brooklyn, Nueva York.

Makerbot Industries vende sus impresoras en Kits diseñados para que la impresora pueda ser construida por cualquier persona con conocimientos técnicos básicos.

Las características principales del modelo de impresora Thing-o-matic empleada en este proyecto son:

- Pertenece a la familia RepStrap.
- Extrusor: MK7, Aguja de 0.4, Plástico de 1.75 mm.
- Electrónica: Arduino Mega 2560 con Electrónica Gen 4.



Figura 8: Imagen de la impresora 3D Thing-o-matic [Fuente: www.impresoras3d.net]

RepRap Air:

Esta basada en el proyecto RepRap, el cual es una iniciativa creada con el propósito de crear una maquina de prototipado rápido libre que sea capaz de auto-replicarse, haciendo un kit de sí misma.

La Impresora 3D Prusa (Fue una revolución de Josef Prusa que consiguió simplificar la construcción de la impresora Mendel que es la segunda y mejorada versión de RepRap) se ha convertido en un estándar en la construcción de una RepRap Air.

El diseño de la RepRap Air se basa en tres principios básicos:

- Simplificar al máximo la fabricación y el montaje (Principio KISS).
- Diseñar las piezas de la forma más económica que sea capaz de cumplir con los requisitos necesarios.
- Un poco de estética.

Sus características principales son:

- Tipo de hotend: J-Head MKV 3mm
- Volumen de impresión: 3910 cm^3 (20×17×11.5cm)
- Extrusor: Jonaskuehling 3mm
- Electrónica: Sanguinololu 1.3b
- Estructura: Metacrilato cortado por laser
- Base caliente: MK2

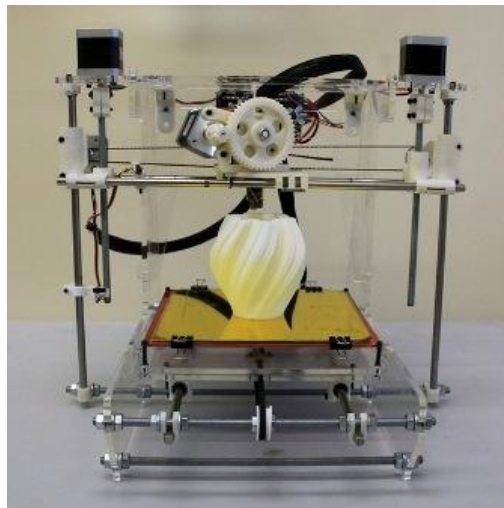


Figura 9: Imagen de la impresora 3D RepRap Air 2 [Fuente: <http://leapto3d.com>]



Capítulo 3

Arduino

3.1 Introducción

3.2 Arduino mega 2560

3.3 Funciones básicas para programación

3.1 INTRODUCCIÓN

Arduino es una plataforma de electrónica libre basado en software y hardware flexibles y de fácil uso para la creación de diferentes prototipos. Al ser open-hardware puede emplearse para realizar cualquier tipo de proyecto sin haber adquirido ninguna licencia.



Figura 10: Imagen del logotipo Arduino [Fuente: www.icsescra.com]

Las placas se pueden montar manualmente o adquirirlas. En este tipo de micro-robots concretamente en este caso un Skybot rediseñado lo mejor es emplear para su fabricación una electrónica libre, para futuras mejoras realizadas por personas ajenas.

El Skybot original utiliza las tarjetas SKYPIC como procesadora y SKY293 como tarjeta de potencia las cuales serán sustituidas por una tarjeta arduino en concreto la Mega 2560 y un shield con el acondicionamiento necesario para los sensores.

3.2 ARDUINO MEGA 2560

El Mega está diseñado para que sea compatible con la mayoría de shields diseñados para otros modelos de tarjetas arduino.

El arduino mega es una placa basada en el microcontrolador ATmega2560. Tiene 54 entradas/salidas digitales, de las cuales 14 proporcionan salida PWM, 16 entradas digitales, 4 UARTS (puertos serie por hardware), un cristal oscilador de 16 MHz, conexión USB, entrada de alimentación, conector ICSP y botón de reset.



Figura 11: Imagen de la placa Arduino Mega 2560 REV3 [Fuente: www.tuxbrain.com]

Características generales:

- Microcontrolador: ATmega2560.
- Voltaje de funcionamiento: 5V.
- Voltaje de entrada recomendado: 7-12V.
- Salida de alimentación de 5V y 3.3V con 50mA.
- Máxima corriente continua para las entradas: 40mA.
- Intensidad en pin 3.3V: 50mA.
- Memoria de programa (Flash Memory) de 256Kb de los cuales 8Kb son usados para el arranque del sistema (bootloader).
- Memoria SRAM de 8Kb para datos y variables del programa.
- Memoria EEPROM para datos y variables no volátiles de 4Kb.
- Velocidad de trabajo del reloj: 16MHz.
- Dimensiones: 100x50 mm.
- Conexión USB.

El Arduino Mega puede ser Alimentado por conexión USB o con una fuente de alimentación externa. Estas pueden ser tanto un transformador como una batería.



Figura 12: Imagen de una batería para Arduino [Fuente: www.vargasbots.com]

Esta tarjeta facilita por sus puertos la comunicación con el ordenador, otro Arduino u otros microcontroladores. También destacar que si es necesario reiniciar la tarjeta en lugar de presionar el botón de reset antes de cargar, es posible reiniciar por software desde el ordenador donde esté conectado.

En cuanto a la protección contra cortocircuitos y sobretensiones el Arduino Mega tiene un multifusible reinicializable que protege la conexión USB del ordenador.

3.3 FUNCIONES BÁSICAS PARA PROGRAMACIÓN

La estructura básica de programación de Arduino es simple y divide la ejecución en dos partes:

- **Setup ():** Constituye la preparación del programa. En esta función se incluye la declaración de variables y se trata de la primera función que se ejecuta en el programa. Se ejecuta una única vez y es empleada para configurar el pinMode (Si un determinado pin digital es de entrada o de salida) e inicializar la comunicación serie.
- **Loop ():** Incluye el código a ser ejecutado continuamente.

Arduino está basado en C y soporta todas las funciones del estándar C y algunas de C++. A continuación se muestra un resumen de las funciones más usadas:

- **pinMode (pin, mode):** Función usada en la función setup () para configurar un pin dado para comportarse como INPUT (entrada) o OUTPUT (salida).

-
- **digitalRead (pin):** Lee el valor desde un pin digital específico. Devuelve un valor HIGH (alto) o LOW (bajo). El pin puede ser especificado con una variable o una constante.
 - **digitalWrite (pin, value):** Introduce un nivel alto (HIGH) o bajo (LOW) en el pin digital especificado.
 - **analogRead (pin):** Lee el valor desde el pin analógico especificado con una resolución de 10 bits. El valor resultante es un entero de 0 a 1023. Los pines analógicos a diferencia de los digitales no necesitan declararse previamente como INPUT u OUTPUT.
 - **analogWrite (pin, value):** Escribe un valor pseudo-analógico usando modulación por ancho de pulso (PWM) en un pin de salida marcado como PWM. Puede especificarse un valor de 0-255. Un valor 0 genera 0V en el pin especificado y 255 genera 5V. Para valores de 0 a 255, el pin alterna rápidamente entre 0V y 5V, cuanto mayor sea el valor, más a menudo el pin se encuentra en HIGH (5V).
 - **delay (ms):** Realiza una pausa en el programa la cantidad de tiempo en milisegundos especificada en el parámetro.
 - **Serial.begin (rate):** Abre un puerto serie y especifica la velocidad de transmisión. La velocidad típica para la comunicación con el ordenador es de 9600.
 - **Serial.println (data):** Imprime datos al puerto serie seguido por un retorno de línea automático. Tiene la misma forma que Serial.print () pero este sin el salto de línea al final. Este comando puede emplearse para realizar la depuración de programas. Para ello pueden mandarse mensajes de depuración y valores de variables por el puerto serie y posteriormente activando desde el software Arduino el “Serial Monitor” se puede observar el contenido del puerto serie.
 - **Serial.read ():** Lee o captura un byte (un carácter) desde el puerto serie. Devuelve un -1 si no hay ningún carácter en el puerto serie.

Capítulo 4

Sistema Sensorial

4.1 Sensor de contacto o Bumpers

4.2 Fotorresistencia

4.3 Sensor CNY70

4.4 Sensor Sharp GP2Y0A2

4.5 Sensor de ultrasonidos

4.1 Sensor de Contacto o Bumpers

Los sensores de contacto son los dispositivos más sencillos que se pueden encontrar, ya que actúan simplemente como un mero interruptor que se activa o desactiva en función de si está en contacto con algún objeto o no.

La principal misión que van a tener este tipo de sensores es evitar posibles daños que pueda tener el robot ante determinados obstáculos.

Existen dos tipos de sensores de contacto:

- **Tipo pulsador:**

Este tipo de sensores consta de un botón, que al ser pulsado para adentro cierra el interruptor interno indicando que hemos chocado con algo.

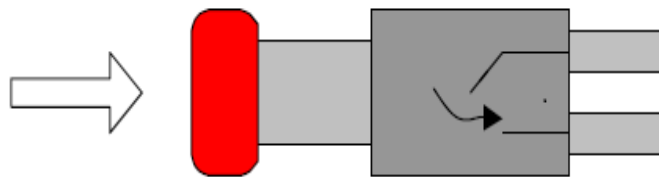


Figura 13: Imagen del funcionamiento de un sensor tipo pulsador

- **Tipo Final de carrera “Bumpers”:**

El funcionamiento de estos sensores es muy simple, es un conmutador que tiene dos posiciones con muelle de retorno hacia la posición de reposo y con una palanca de accionamiento con mayor o menor longitud según el tipo de Bumper elegido.

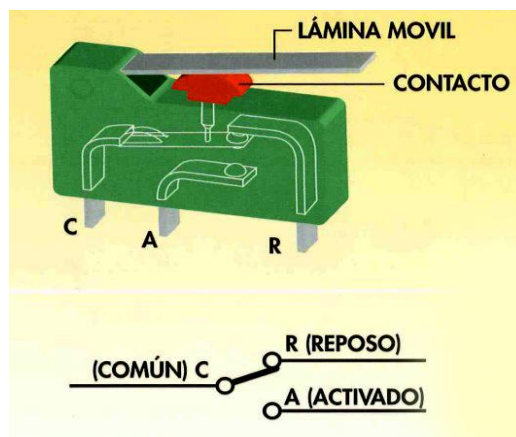


Figura 14: Imagen de un sensor bumper [Fuente: www.comohacerturobot.com]

Como se puede ver en la imagen en estado de reposo la patilla común (c) y la de reposo (R) están en contacto permanente hasta que se presiona la palanca, entonces el contacto pasa de la posición de reposo a la de activo (A), cuando ocurre esto podemos escuchar un pequeño clic.

El principal inconveniente que tienen estos sensores es que no se pueden emplear en robots de alta velocidad ya que cuando detectan el obstáculo ya están encima y no da tiempo a frenar el robot.

Estos sensores pueden implementarse en el sistema de dos formas:

- Como otro sensor más que va al sistema del robot.
- Conectándolo directamente al circuito de control de los motores de forma que se pueda parar el robot ante cualquier obstáculo.

En este micro-robot el sensor de contacto empleado es de tipo Bumper y la implementación es que ante cualquier obstáculo el micro-robot se parara. Pudiendo modificar que se pare por completo o no mediante la programación.

4.2 Fotorresistencia:

Se las conoce también como LDR (Light Dependent Resistor), son resistencias cuyo valor varía de acuerdo al nivel de luz al que están expuestas.

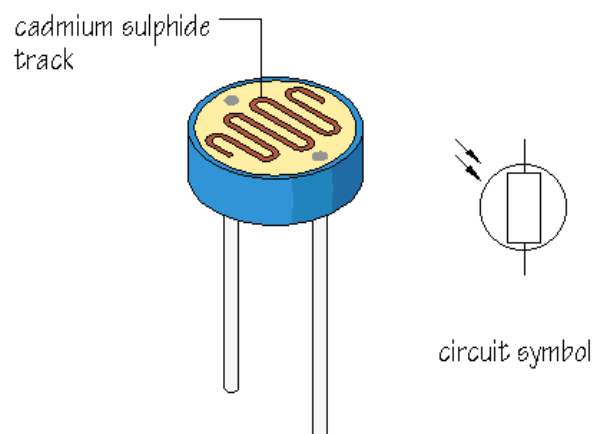


Figura 15: Imagen de una fotorresistencia [Fuente: www.capacitacion-electronic84.blogspot.com]

El valor de la resistencia eléctrica es bajo cuando hay luz incidiendo en el (hasta los 50 ohmios) y muy alto cuando está a oscuras (varios mega ohmios).

El funcionamiento de la LDR en el hecho de que la resistencia eléctrica de una película de sulfuro de cadmio varía en función de la intensidad de luz que incide sobre la superficie de la película.

La relación entre el valor de la resistencia y la iluminación, se puede expresar aproximadamente mediante la siguiente expresión:

$$R = A \times L^{\alpha}$$

Donde R es el valor de la resistencia (Ω), L es la iluminación (lux) y A y α son constantes (α depende del proceso de fabricación y del material utilizado y varía de 0.7 a 0.9).

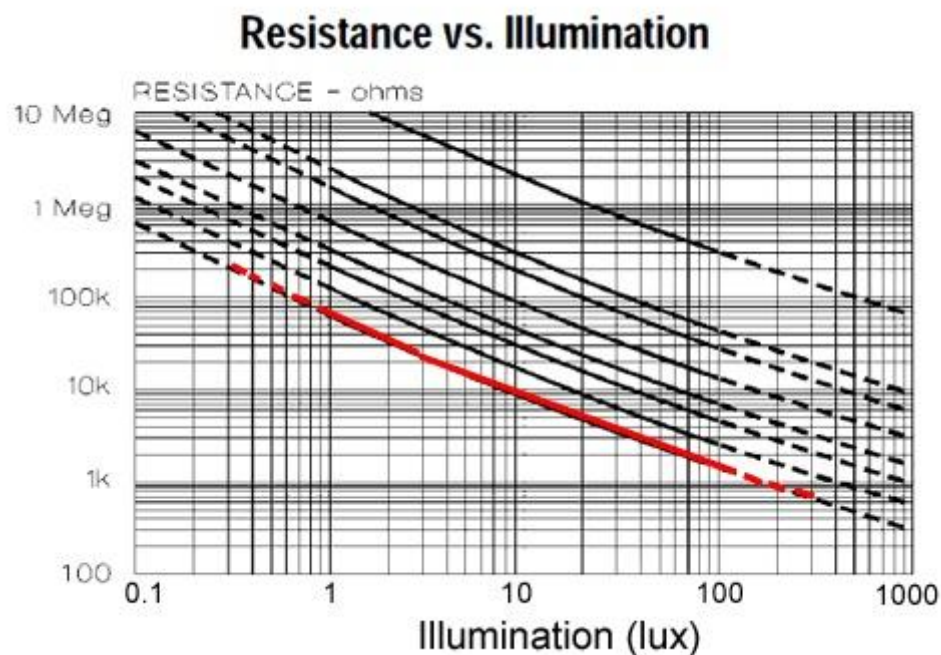


Figura 16: Grafica de LDR representando la resistencia frente a la Iluminación.

La LDR a pesar de ser un dispositivo sensible, económico y fácil de conseguir tiene un inconveniente y es que reaccionan lentamente, esto limita su uso, ya que no se pueden emplear en aplicaciones en las que la señal luminosa varía con rapidez.

4.3 Sensor CNY70:

El CNY70 es un sensor de infrarrojos de corto alcance basado en un emisor de luz y un receptor, ambos señalan en la misma dirección.

El emisor es un fotodiodo que se encarga de emitir luz cuya intensidad dependerá de la resistencia que se ponga en serie con él y la tensión a la que se alimente.

El receptor es un fototransistor que se encarga de recibir la luz emitida por el fotodiodo cuando se refleja sobre alguna superficie.

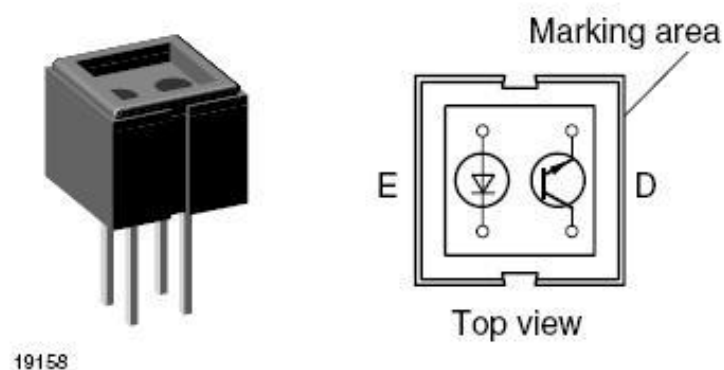


Figura 17: Imagen de un sensor CNY70 [Fuente: www.datasheetdir.com]

Para lo que más se emplea este sensor en la Robótica es como seguidor de líneas ya que se puede detectar diferentes superficies en función de la luz que reflejen, y así es como se sigue una línea negra sobre un fondo blanco, o viceversa.

Según el montaje que se emplee el sensor CNY70 devolverá por la pata de salida un voltaje relacionado con la cantidad de rayo reflejado por el objeto. Se leerá del emisor un "1" cuando se refleje luz y un "0" cuando no se refleje, o viceversa según como estén colocadas las resistencias.

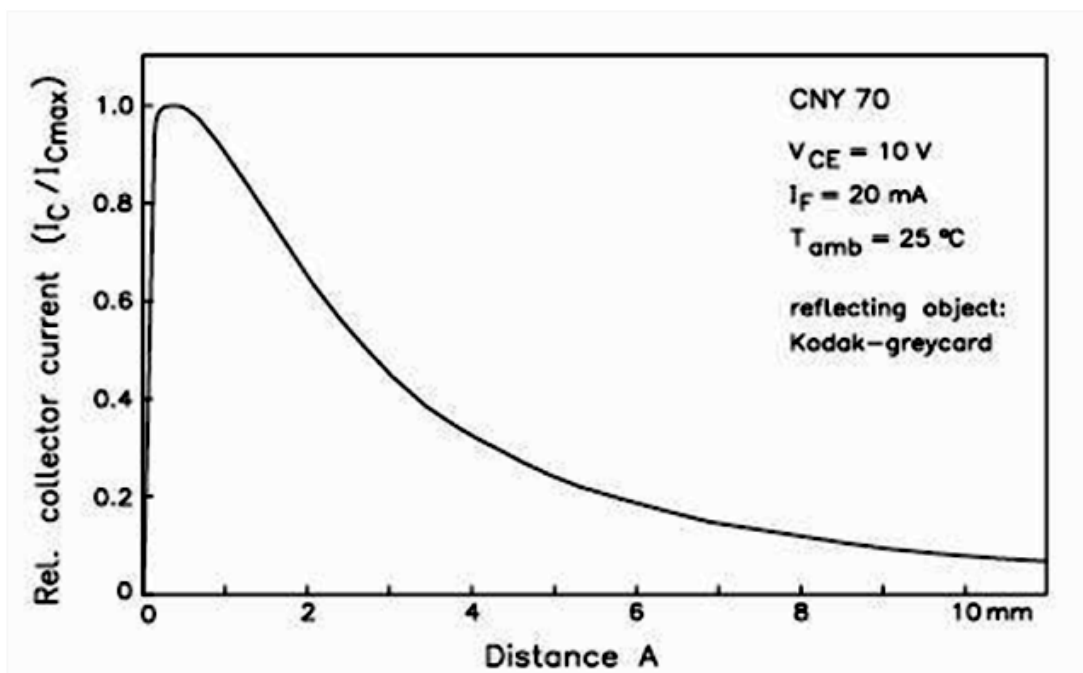


Figura 18: Grafica de sensor CNY70 representando la intensidad frente a la distancia.

El principal inconveniente que tiene este sensor es que se debe situar muy cercano al suelo ya que como se puede observar en la grafica a partir de 6mm de distancia la intensidad a caído más del 80% del valor inicial.

4.4 Sensor Sharp GP2Y0A2:

Es un sensor de infrarrojos que proporciona continuamente la distancia como una tensión analógica. Para este modelo esta varia 2.8V a 20cm hasta 0.4 a 150cm con una fuente de alimentación entre 4.5 y 5.5 VDC.



Figura 19: Imagen de un sensor Sharp GP2Y0A2 [Fuente: www.olimex.cl]

A diferencia de otros modelos el encapsulado de este presenta una distancia entre la lente y el sensor con el fin de aumentar el rango de trabajo. Estos sensores utilizan un conector especial en concreto es un JST (Japanese Solderless Terminal) como el que se ve en la siguiente figura.



Figura 20: Imagen de un conector JST [Fuente: www.picaxe.biz]

Se puede observar que el conector JST es de 3 vías, 2 para la alimentación y una para la salida, cuyo valor se proporciona continuamente y se actualiza cada 39ms.

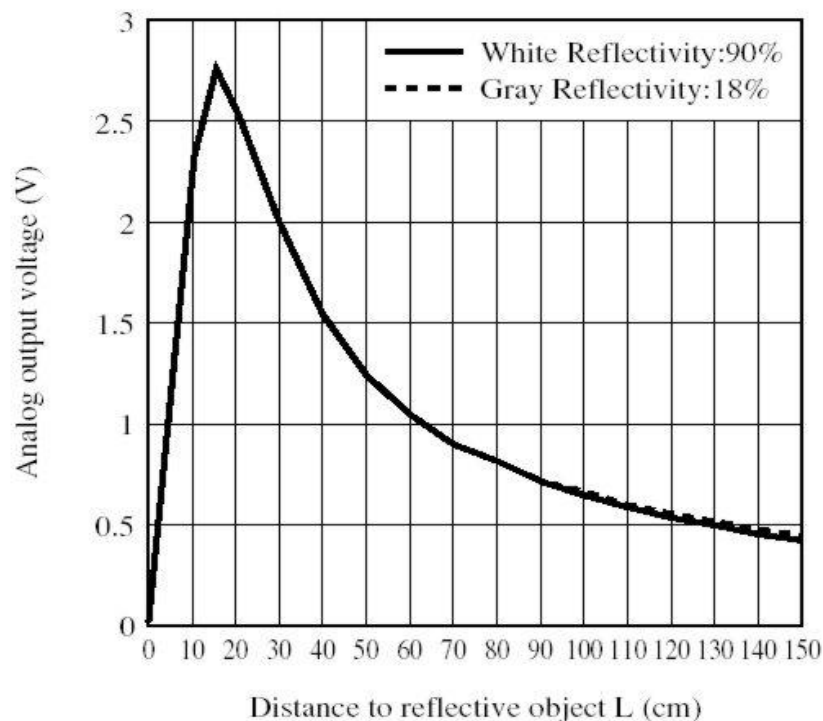


Figura 21: Grafica del sensor Sharp GP2Y0A2 que representa la variación del voltaje frente a la distancia.

4.5 Sensor de ultrasonidos:

Son sensores que se emplean para medir la proximidad a un objeto sin necesidad de contacto. En este proyecto se emplea un ultrasonido tipo sonar, exactamente el SRF05, este sensor es una evolución del clásico SRF04. La diferencia con este es que tiene dos modos de funcionamiento, el modo estándar se comporta igual que el SFR04 con la diferencia de que el rango de trabajo aumenta de 3 a 4 metros y el nuevo modo que emplea un solo pin para controlar el sensor y hacer la lectura de la medida.



Figura 22: Imagen de un sensor SRF05 [Fuente: www.solarbotics.com]

Modo 1:

En este modo se emplean terminales independientes para la señal de activación y para la señal de retorno del eco. Para utilizar este modo se deberá de dejar el pin de modo sin conectar, como se puede ver en la siguiente imagen.

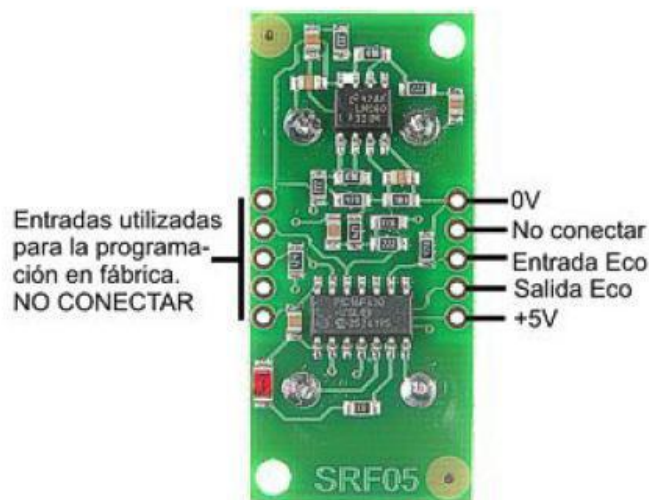


Figura 23: Imagen de un sensor SRF05 en modo 1 [Fuente: www.jsadalid.wordpress.com]

Este modo es el más fácil de programar ya que tienes terminales independientes para la entrada y para la salida, pero esto implica la necesidad de conectar dos cables desde arduino hasta el sensor.

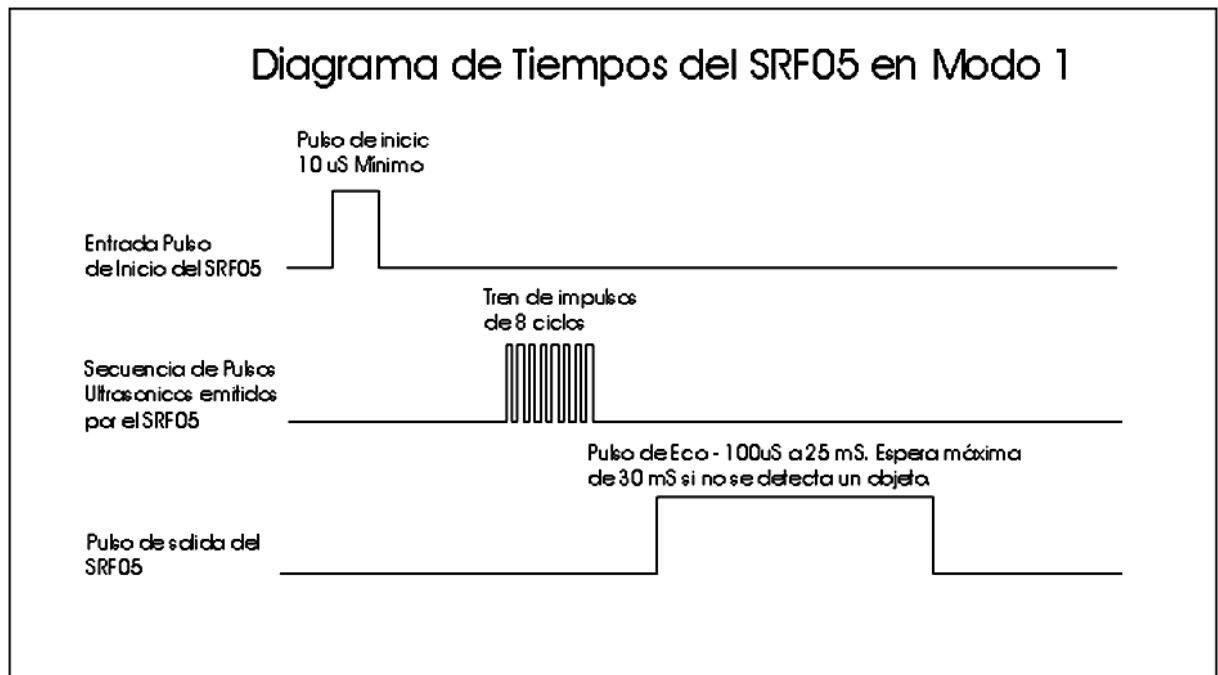


Figura 24: Diagrama de tiempos del SRF05 modo 1

Como se puede ver Por uno de ellos se le envía el pulso de disparo (trigger) y por el otro se recibe el pulso de eco (echo) cuya amplitud es directamente proporcional a la distancia a la que ha sido detectado el obstáculo interpuesto.

Modo 2:

Este modo utiliza un único pin para las señales de activación y eco, está diseñado para reducir el número de pines en los microcontroladores.

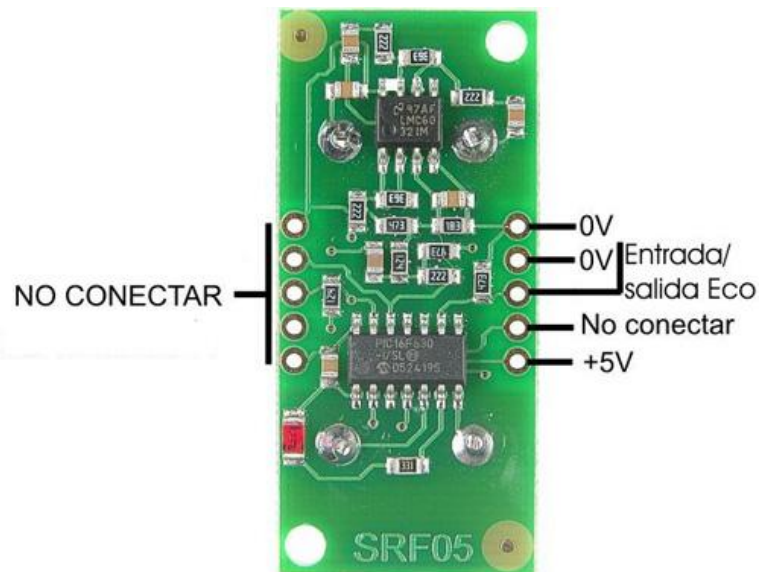


Figura 25: Imagen de un sensor SRF05 en modo 2 [Fuente: www.muchotrasto.com]

Para medir la distancia con arduino se utiliza la función “PulseIn”, la cual lee un estado HIGH o LOW sobre la terminal apropiada. En este modo es necesario conectar el terminal de modo a la tierra del arduino.

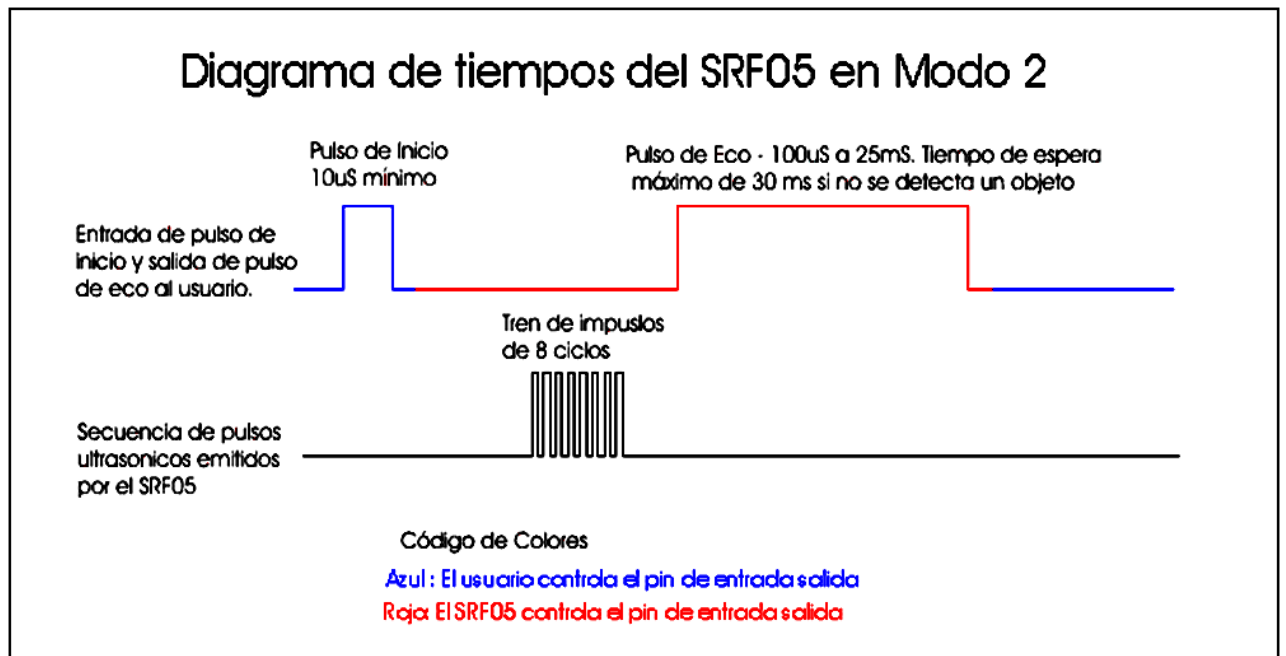


Figura 26: Diagrama de tiempos del SRF05 modo 2

El SRF05 trabaja a una frecuencia de 40 kHz y a una tensión de 5 voltios. Su principal aplicación es como detector de obstáculos en robots con navegación autónoma. Su haz se muestra en la siguiente figura.

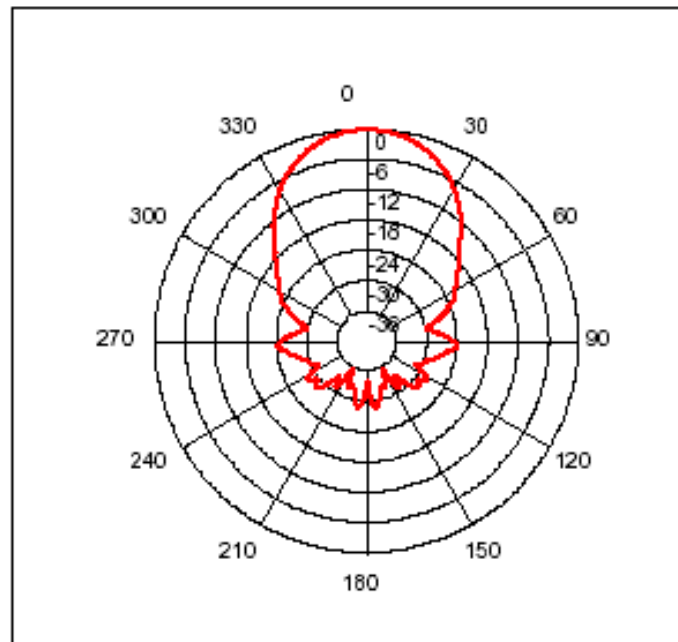


Figura 27: Diagrama del haz del SRF05



Capítulo 5

Software

5.1 Openscad

5.2 Arduino software

5.3 Eagle

5.4 Slic3r y Pronterface

5.1 OPENS CAD

Es una herramienta de diseño para la creación de objetos tridimensionales. Es un software gratuito y está disponible para Linux/UNIX, Windows y Mac OS X.

Este software no es interactivo sino que permite crear piezas a partir de un script que al ejecutarse se muestra la pieza 3D en pantalla. Esto proporciona al diseñador una mayor flexibilidad a la hora de realizar algún cambio en la pieza ya que solo tiene que modificar la programación.

OpenScad proporciona dos técnicas de modelado principales: En primer lugar se tiene la geometría sólida constructiva (CSG) y la segunda es la extrusión de contornos 2D. Para este proyecto se ha utilizado la primera técnica.

Una de las razones por las que se ha escogido este software, aparte de que es libre y multiplataforma, es que permite leer y crear modelos tridimensionales en el formato STL, el cual es necesario para posteriormente imprimirlo en una impresora 3D.

A continuación se muestra una imagen con la interface del usuario.

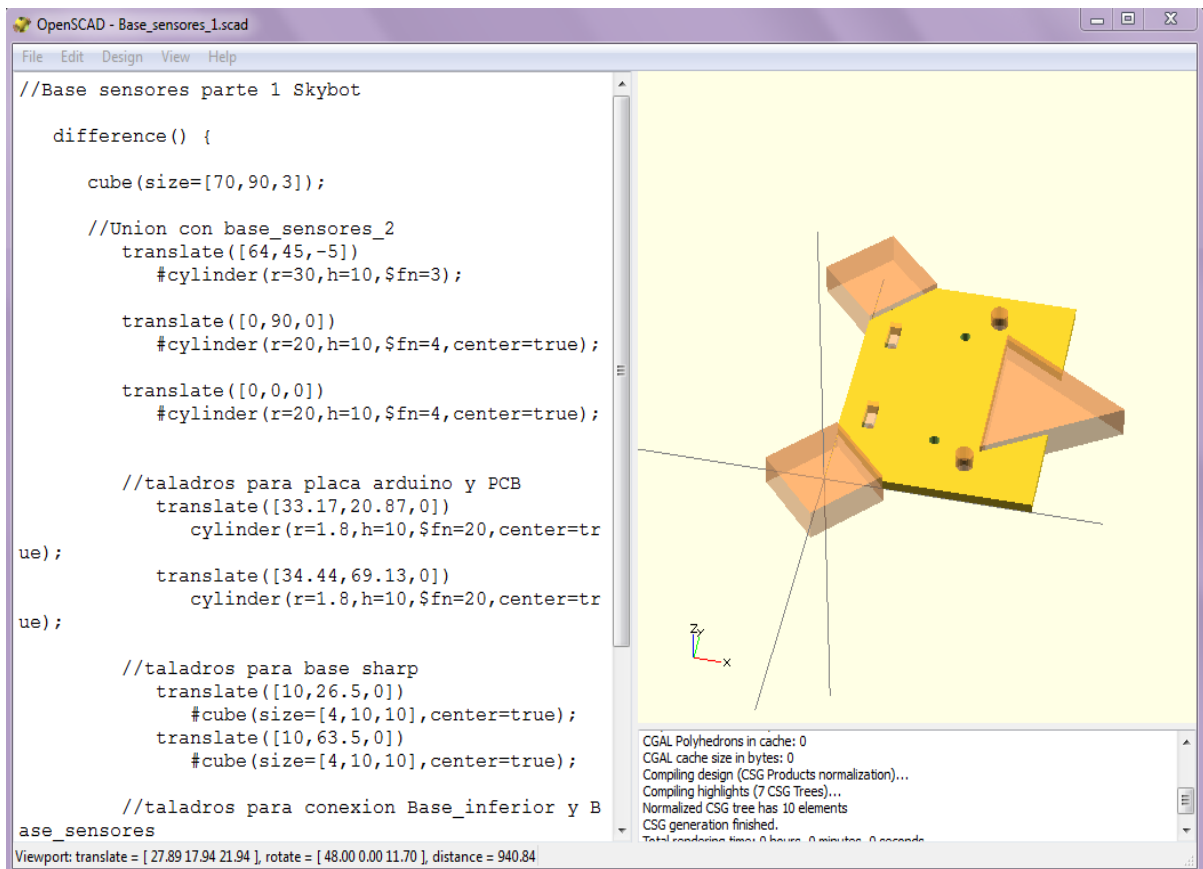


Figura 28: Imagen de la interface OpenScad [Fuente: Elaboración propia]

5.2 ARDUINO SOFTWARE

Es un entorno de código abierto el cual permite una fácil escritura del código y cargarlo a la placa E/S. Funciona en Windows, Mac OS X y Linux. El entorno está escrito en java y basado en Processing, avr-gcc y otros programas también de código abierto.

El código fuente del software arduino puede ser navegado online o descargado. La propia página de la empresa proporciona distintas versiones de este software pudiéndolas descargar fácilmente.

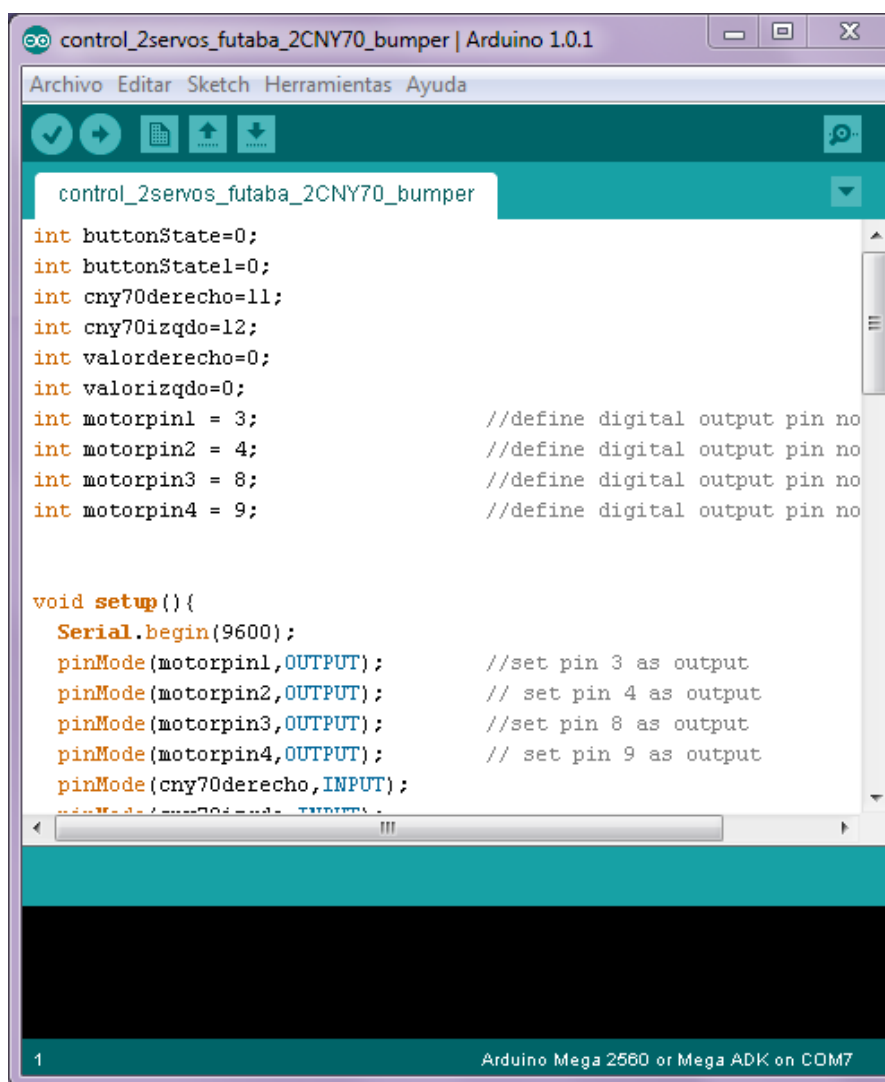


Figura 29: Imagen de la interface del arduino software [Fuente: Elaboración propia]

5.3 EAGLE

Eagle es lo que se conoce como una herramienta CAD (diseño asistido por computador) para la elaboración de circuitos impresos de baja/mediana complejidad. Funciona en Linux, Mac y Windows.

Este software no es libre pertenece al grupo Cadsoft Computer GmbH y Cadsoft Inc., los cuales son subsidiarias de Premier Farnell pic. Uno de los distribuidores líderes en el mundo de componentes electrónicos, productos y servicios. Sin embargo, también es posible usar este programa con una licencia gratuita, aunque con algunas restricciones como limitación del tamaño de la tarjeta a diseñar.

Eagle permite hacer la conexión de los diferentes componentes de un circuito a través del editor de esquemáticos, hacer el trazado de las rutas en el circuito impreso mediante un editor de pcbs y generar los archivos de fabricación. Es decir que contiene tres módulos: El Esquema Editor, el Layout Editor y Autorouter.

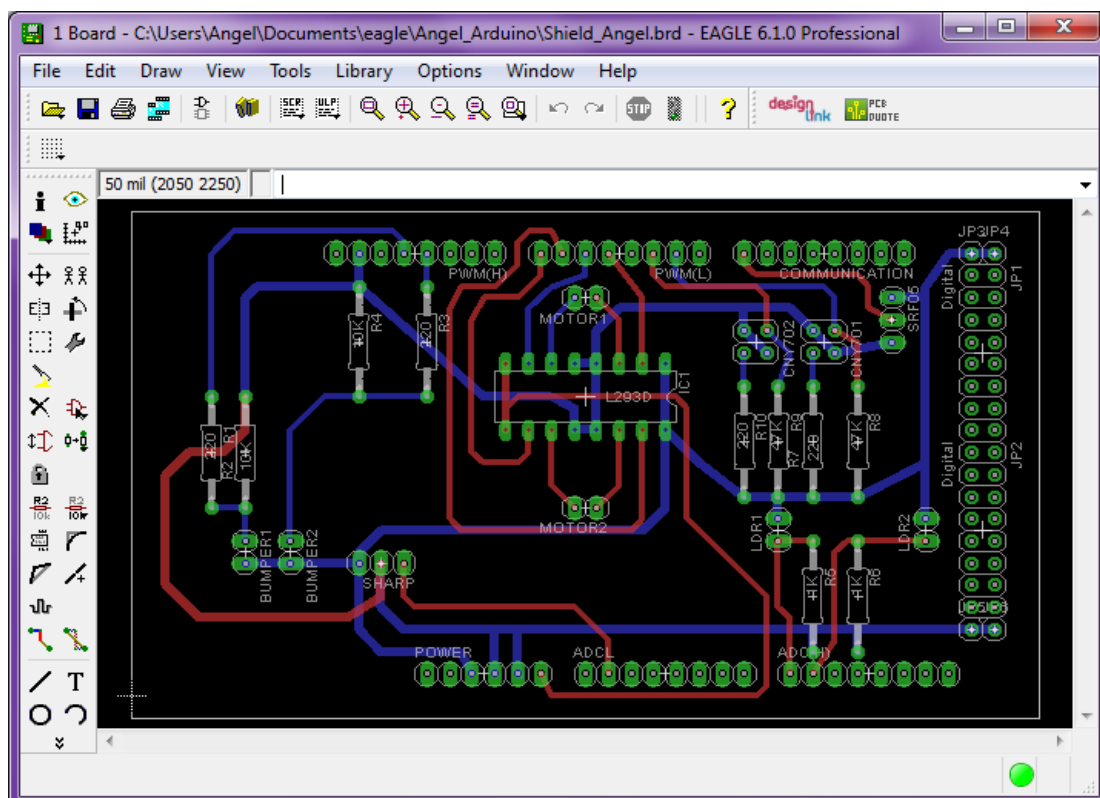


Figura 30: Imagen de la interface de Eagle del Layout Editor [Fuente: Elaboración propia]

Ya que como uno de los aspectos de este proyecto es diseñar una shield para la placa arduino este programa contiene componentes de arduino como pines etc. Este motivo hizo que en este caso se emplease este software en vez de otros como podía ser el Kicad.

5.4 SLIC3R Y PRONTERFACE

Slic3r:

Slic3r es una herramienta que se emplea para convertir un modelo digital en 3D en las instrucciones de impresión para la impresora 3D.

Su función consiste en trocear en capas horizontales la pieza tridimensional que se va a imprimir y genera las trayectorias que deberá seguir el hot-end (dispositivo que se encarga de empujar un hilo de plástico una vez que lo ha fundido) para poder imprimir la pieza calculando la cantidad de material a extruir. También generara las coordenadas y movimientos que tendrán que realizar los motores de cada eje para que el hot-end pueda realizar las trayectorias generadas.

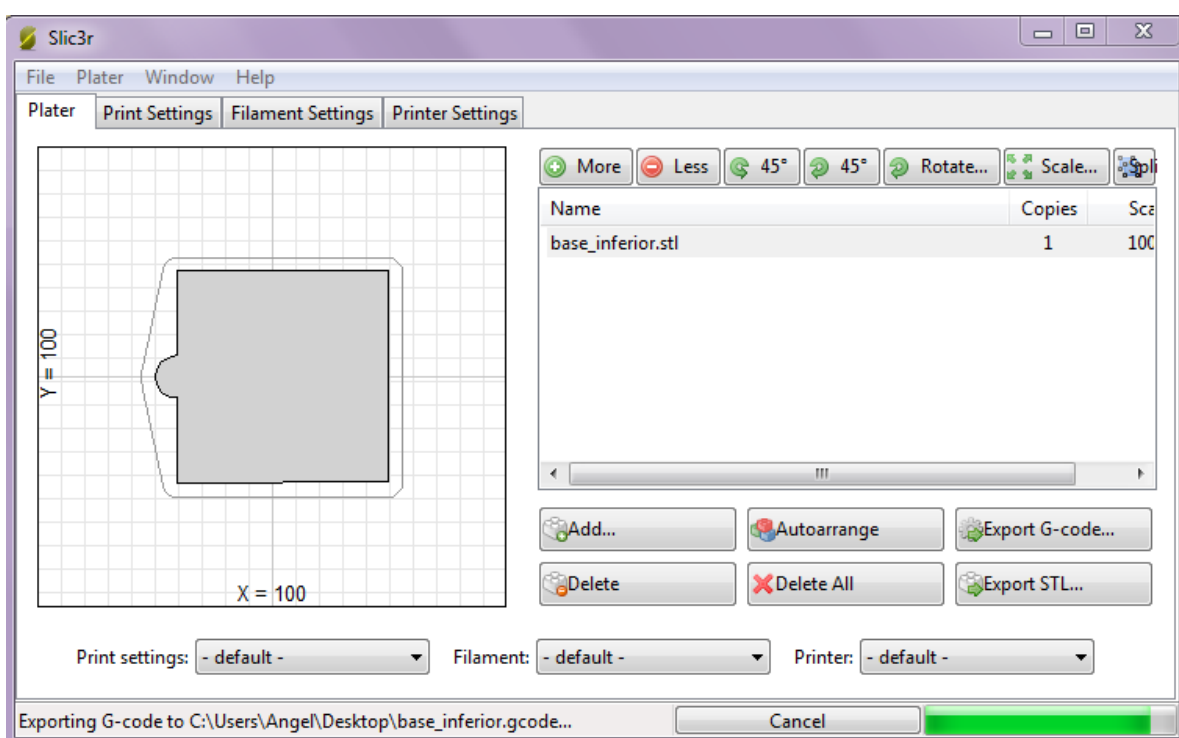


Figura 31: Imagen de la interface de Skeinforge [Fuente: Elaboración propia]

Pronterface:

Pronterface es un software que pertenece al proyecto PrintRun, el cual es un conjunto de aplicaciones de envío de código G y consiste en printcore, pronsole, pronterface y una pequeña colección de scripts útiles.

Pronterface en concreto tiene funciones de envío de código G con interfaz grafica de usuario. Este permite controlar la impresora 3D desde el ordenador e imprimir. Con este programa se controlara la temperatura, el tiempo de impresión, se podrá manejar manualmente los ejes, parar o pausar la impresión entre otras muchas funciones.

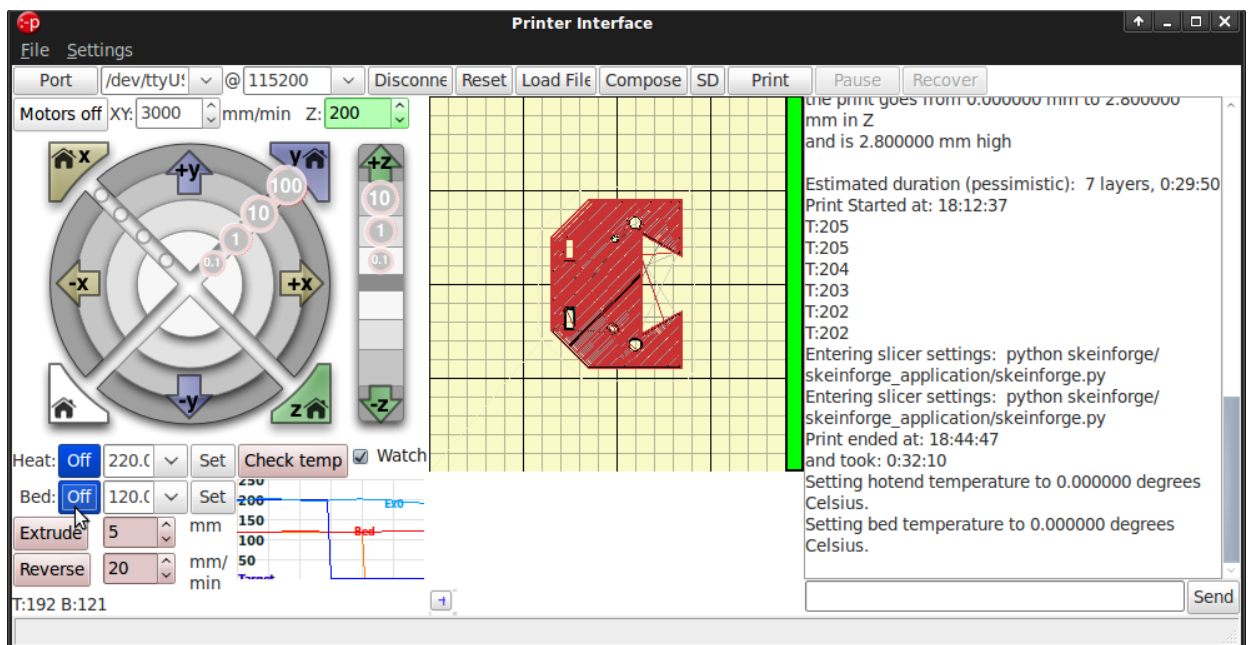


Figura 32: Imagen de la interface de Pronterface [Fuente: Elaboración propia]



Capítulo 6

Sistema Locomotor

6.1 Elección de los motores

6.2 Control de los motores

6.1 ELECCIÓN DE LOS MOTORES

Como el robot es móvil es necesario disponer de motores que hagan girar las ruedas. Se pueden usar dos tipos de motores, paso a paso o de corriente continua. Nos centraremos en los segundos por que son más sencillos y económicos.

Los motores de corriente continua son muy fáciles de controlar. Disponen de dos cables donde se aplica un voltaje. El funcionamiento se basa en la interacción entre el campo magnético del imán permanente y el generado por las bobinas, ya sea una atracción o una repulsión hacen que el eje del motor comience su movimiento. En cuanto al sentido del giro depende de la polaridad aplicada.

El problema de los motores de corriente continua es que giran muy rápido y tienen poca fuerza, lo que los hace poco adecuados para la construcción de robots móviles destinados al aprendizaje ya que queremos que se muevan sobre una superficie pequeña. Por lo que es necesario emplear una caja reductora con un eje de salida que gira más lentamente pero con más fuerza.

Hacer una caja reductora es complicado y las que se pueden comprar son caras por lo que hemos decidido emplear servomotores trucados.

Un servomotor es un tipo de motor que permite controlar la posición en la que se ubica y mantenerse estable en esa posición. Generalmente este tipo de motores tiene un ángulo de giro limitado, que suele ser 180°.

Internamente un servomotor está formado por los siguientes elementos:

- Un motor de corriente continúa.
- Un conjunto de engranajes (caja reductora).
- Un circuito de control.

Tras esta breve descripción de lo que es un servomotor lo que queremos es transformar el servomotor en un motor de corriente continua con reductora por los problemas que antes se ha mencionado.

Para ello se eliminara la electrónica del servomotor ya que el control del motor lo vamos a hacer desde el microcontrolador, que posteriormente se tratara en otra sección. También se deberá de eliminar la limitación de girar solamente 180°. Los pasos para trucar un servomotor se pueden encontrar fácilmente en internet ya que hay numerosas páginas donde lo indican.

Se emplearan los servos Futaba 3003 trucados, las características principales son:



- Velocidad: 260 grados/seg.
- Par de salida: 3.2 Kg-cm (0.314 N.m).
- Dimensiones: 40.4 x 19.8 x 36 mm.
- Peso: 37.2 gramos.
- Frec. PWM: 50 Hz.

Figura 33: Imagen del servo Futaba 3003 trucado [Fuente <http://arantxa.ii.uam.es/~gdrivera>]

6.2 CONTROL DE LOS MOTORES

El control directo de un motor de corriente continua desde un microcontrolador, en este caso es la arduino mega 2560, no es técnicamente posible debido a su carácter inductivo. Esto se debe a que los motores desde su puesta en marcha hasta alcanzar su velocidad normal de funcionamiento consumen una intensidad variable que las salidas de un microcontrolador no están en condiciones de suministrar de manera fiable.

Por ello para controlar los motores se ha empleado el driver L293, que puede presentar distintas variantes. En este caso se ha empleado el L293DNE debido a que viene incorporado con 4 diodos de protección que evitan los daños producidos por los picos de voltaje que pueden producir los motores, así no tenemos que incluirlos en nuestro circuito impreso y nos ahorramos espacio entre otras cosas. A continuación se detallara como es este driver y el conexionado con los motores y la placa Arduino.

INTEGRADO L293DNE

El L293D es un puente H (Circuito electrónico que permite a un motor eléctrico DC girar en ambos sentidos, avance y retroceso) tipo IC que incluye cuatro circuitos para manejar cargas de potencia media, en especial pequeños motores y cargas inductivas, con la capacidad de controlar corriente hasta 600 mA en cada circuito y una tensión entre 4,5 a 36 V.



Figura 34: Imagen de un integrado L293DNE [Fuente www.archiwumallegro.pl]

El integrado permite formar dos puentes H completos con los que se puede realizar el manejo de dos motores. En la siguiente imagen vamos a mostrar el patillaje de este integrado.

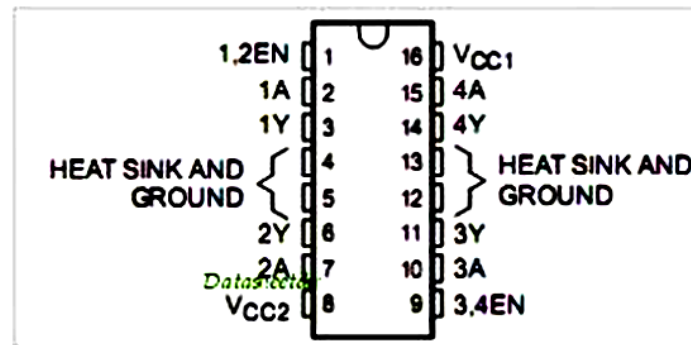


Figura 35: Imagen del patillaje de un integrado L293DNE [Fuente www.datasheetdir.com]

CONEXIONADO

Para realizar la conexión se ha seguido el patillaje que nos indica el datasheet del integrado L293DNE. A la placa Arduino se conecta a los pines digitales 4, 5, 6, 7 y a parte de emplear la alimentación de 5V se emplea la VIN de 3.3V.

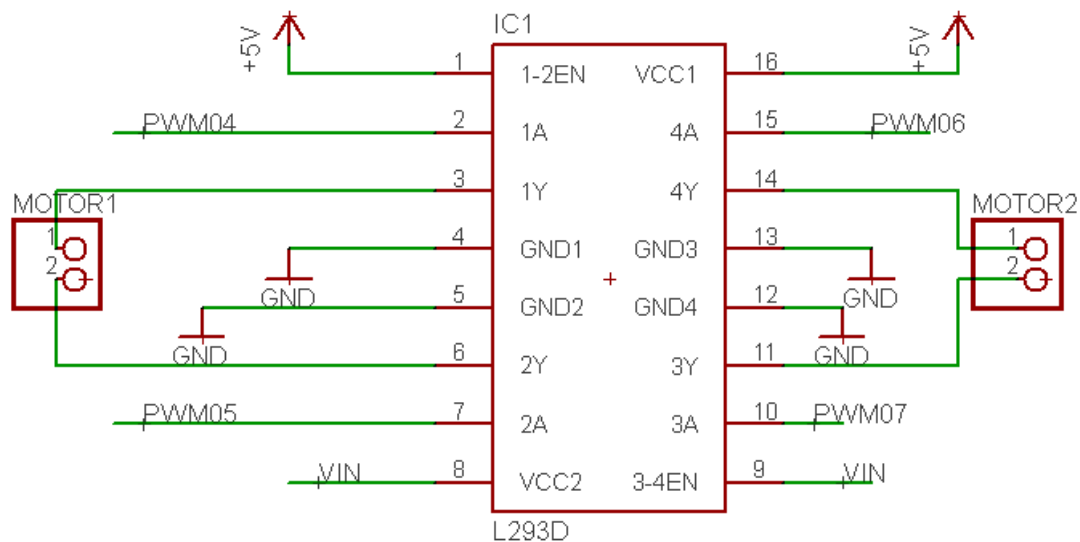


Figura 36: Imagen conexionado de los motores al L293D y a placa Arduino
[Fuente: Elaboración propia]



Capítulo 7

Diseño Electrónico

7.1 Introducción

7.2 Acondicionamiento de los sensores

7.3 Diseño del circuito impreso

7.1 INTRODUCCIÓN

A pesar de que este micro-robot está basado en el Skybot, el sistema electrónico es totalmente diferente.

El Skybot emplea las tarjetas Skypic como procesador y sky293 para los sensores y la etapa de potencia. Ambas se unen formando una torre encima del chasis. La tarjeta Skypic lleva un microprocesador PIC16F876A el cual se programara con aplicaciones que existen.

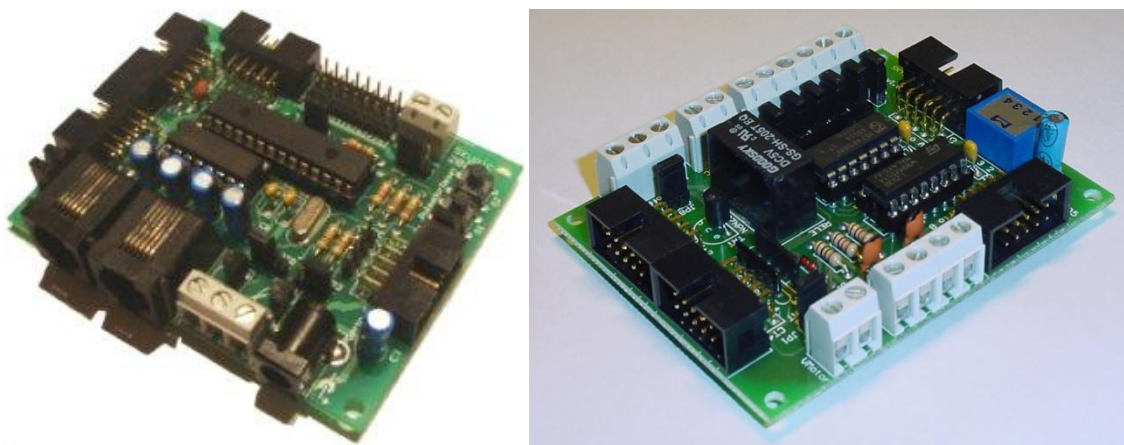


Figura 37: Imagen de las tarjetas Skypic (Izquierda) y Sky293 (derecha) [Fuente: www.learobotics.com/wiki]

Para el micro-robot de este proyecto se emplea la Tarjeta Arduino Mega 2560 para la programación de todo el sistema sensorial y motor y un circuito impreso en el que se encuentra el acondicionamiento de cada uno de los sensores y el driver L293D para controlar los dos motores.

A continuación se mostrara el acondicionamiento necesario para cada sensor utilizado, el diseño del circuito impreso mostrando el conexionado que se ha realizado con imágenes de Eagle. En el capítulo 8 se mostrara como se ha programado cada sensor ya que algunos necesitaban previamente que se les calibre.

7.2 ACONDICIONAMIENTO DE LOS SENSORES

A continuación se detallara como hay que acondicionar y conectar los sensores que se emplean.

7.2.1 Sensor de contacto o Bumper:

Este micro-robot lleva dos sensores Bumper cada uno de ellos se deben de acondicionar con una resistencia de $220\ \Omega$ y $10\ K\Omega$. Estas resistencias serán de $\frac{1}{4}$ de Watio y tendrán un 5% de tolerancia.

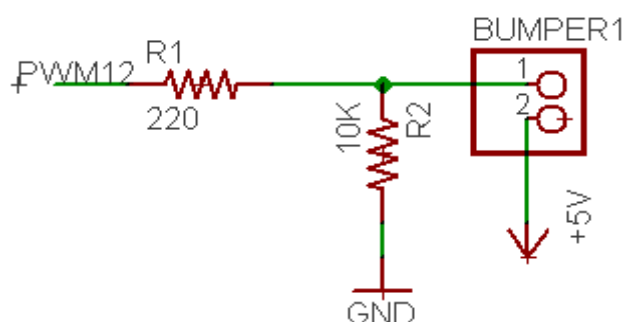


Figura 38: Imagen conexionado de un Bumper [Fuente: Elaboración propia]

En esta imagen se muestra el acondicionamiento y el conexionado tanto de los componentes como a la placa arduino. Las resistencias están conectadas en serie y a la placa arduino se conectara en el pin 12 y pin 11 digital ya que devolverá un cero o un uno si se pulsa o no.

7.2.2 Fotorresistencia o LDR:

En este caso se utilizaran 2 fotorresistencias las cuales se acondicionaran con una resistencia de $1K\Omega$ de $\frac{1}{4}$ de Watio con un 5% de tolerancia. El circuito constara de un divisor de tensión el cual dará una tensión de la salida que cambia con la iluminación.

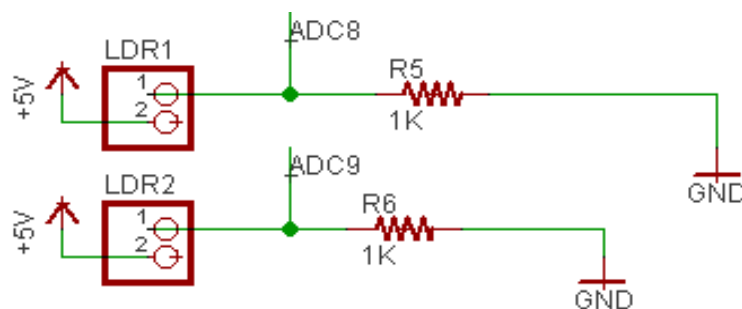


Figura 39: Imagen conexionado de las LDR [Fuente: Elaboración propia]

Como podemos ver en esta imagen se muestra el circuito comentado anteriormente. Las LDR a la placa arduino se conectan al pin 8 y 9 analógicamente ya que se recibirá los valores comprendidos entre el 0 y el 1023 según la cantidad de luz que se refleje en las LDR.

7.2.3 Sensor CNY70:

Para que este micro-robot siga las líneas se emplearan dos sensores CNY70, estos para poder conectarlos a arduino hay que acondicionarlos como los sensores vistos anteriormente. En este caso se utilizara una resistencia de 220Ω y otra de $47K\Omega$. Ambas serán de $\frac{1}{4}$ de Watio y tendrán 5% de tolerancia.

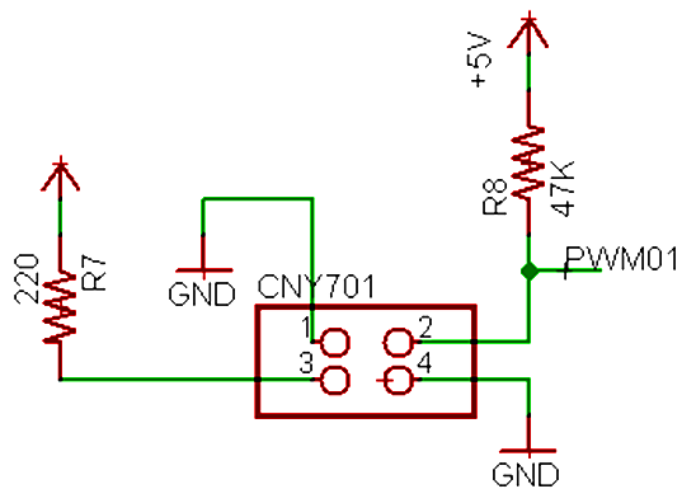


Figura 40: Imagen conexionado de un CNY70 [Fuente: Elaboración propia]

El circuito de la imagen muestra como se conectan las resistencias al sensor y este a la placa arduino. Cada sensor CNY70 se conectara a dos pines digitales, en concreto al 1 y al 2 en el primer prototipo y por problemas de programación en el resto se conecta al 2 y 3. Se conectan digitalmente ya que se recibirá un uno o un cero cuando se refleje o no la luz.

7.2.4 Sensor Sharp:

El sensor Sharp no necesita ningún acondicionamiento especial se conecta directamente a la corriente, a tierra y al pin 1 analógico. Se recibirán diferentes valores comprendidos entre el 0 y el 1023 dependiendo de la distancia a la que se encuentre el objeto u obstáculo.

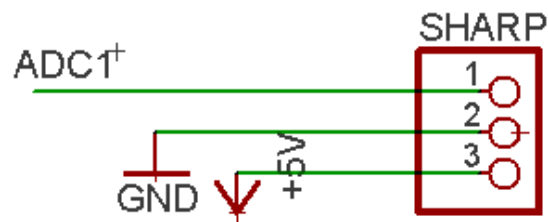


Figura 41: Imagen conexionado del sensor sharp [Fuente: Elaboración propia]

7.2.5 Sensor de Ultrasonidos:

El sensor ultrasonido no necesita ningún tipo de acondicionamiento se conecta directamente a la fuente de corriente, a tierra y al pin 14 digital. En el capítulo 5 ya se explicó los modos de funcionamiento de este sensor y que en este caso se ha empleado el modo 2.

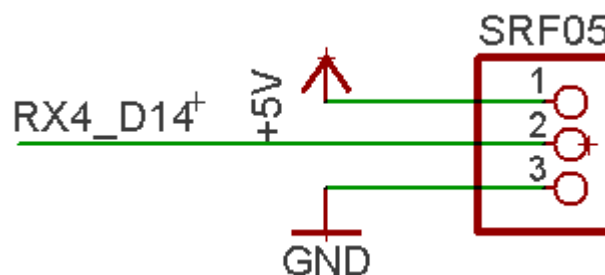


Figura 42: Imagen conexionado del sensor ultrasonido [Fuente: Elaboración propia]

7.3 DISEÑO DEL CIRCUITO IMPRESO

Como se expuso para diseñar el circuito impreso se barajaron varios programas de diseño como el Kitcad, Orcad... Finalmente se decidió emplear el software Eagle ya que aparte de que tiene una versión libre con limitaciones, incluye muchos componentes de Arduino como pines por lo que facilita su diseño.

Antes de realizar el esquemático y pasar a fabricar el circuito impreso se probó a montar el circuito en una placa protoboard y hacer pruebas.

Pasos y consideraciones durante el diseño:

Primeramente una vez que se tiene claro cómo debe de ser la placa y el circuito se realiza el diseño esquemático (Ver anexo 5) donde se dibuja el circuito con las

conexiones y los componentes elegidos correctamente desde las librerías del software en este caso de Eagle.

Posteriormente se pasa a realizar el diseño Board es decir colocar los componentes en la placa y rutearla. En este caso se ha ruteado manualmente aunque se tarde más. Es aconsejable que el grosor de la vía de alimentación y tierra sea mayor, como que los conectores de los sensores estén cerca de su acondicionamiento (resistencias).

Por último una vez que se tiene colocado todos los componentes y ruteada la placa se pasa a generar los archivos gerber y excellon para enviar al fabricante que realizara la placa impresa. Estos archivos se generan con el CAM Processor desde el Board de Eagle. Serán necesarios generar los siguientes archivos eligiendo en cada caso las capas necesarias desde Nr Layer.

- Top Copper (Cara superior), Bottom Copper (Cara inferior).
- Top Silkscreen (Serigrafía superior), Bottom Silkscreen (Serigrafía inferior).
- Top soldermask (Mascara de soldadura superior), Bottom soldermask (Mascara de soldadura inferior).
- Mill layer (Capa de fresado) y Drill file (Fichero de taladrado).

A continuación se muestra el diseño de la placa antes de mandarla a fabricar.

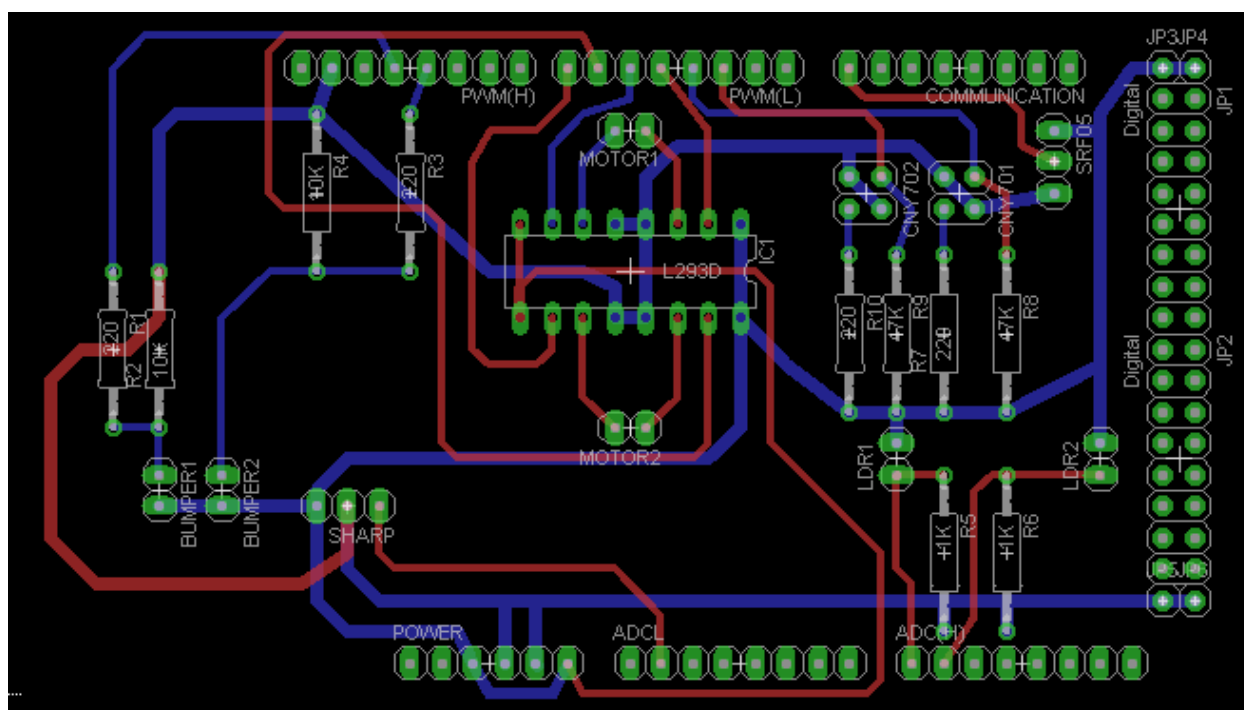


Figura 43: Imagen del diseño de la placa impresa [Fuente: Elaboración propia]

Descripción de la placa impresa:

El circuito impreso es una shield por lo que tiene los mismos pines que la placa Arduino ya que va montada arriba de esta. En la shield irán conectados los sensores y los motores con su acondicionamiento.

Las dimensiones de la placa son 102X58mm, un poco más grande que la placa Arduino sobre la que va colocada. A continuación se muestra la placa impresa.

Para soldar los componentes se ha empleado estaño y un soldador eléctrico con punta fina. Nos hemos ayudado con un desoldador cuando se ponía exceso de estaño.

Los componentes que hemos soldado son pines macho para conectar la placa impresa a Arduino y poder conectar los sensores mediante conectores, las resistencias que acondicionan los sensores y un zócalo donde ira situado el L293D.

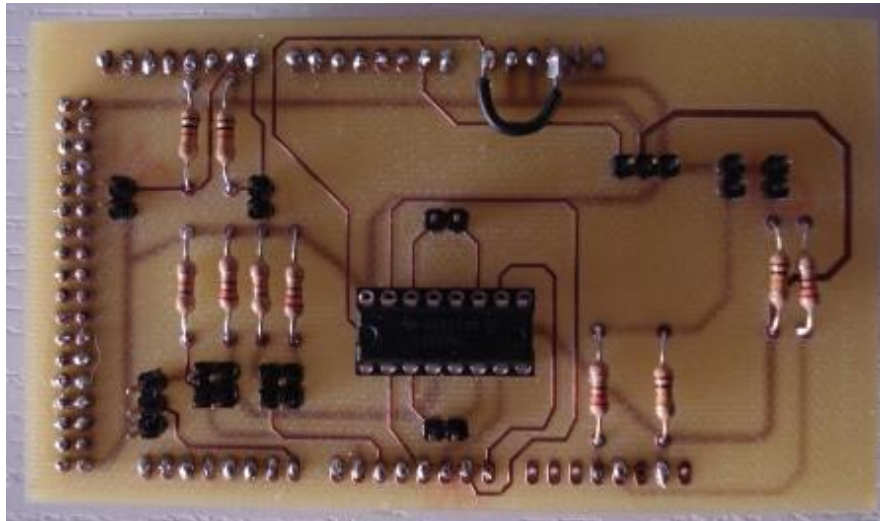


Figura 44: Imagen palca impresa con componentes [Fuente: Elaboración propia]

Para conectar tanto los sensores como los motores a la placa impresa en lugar de soldar los cables directamente a la placa por si algún sensor fallase poderlo cambiar con facilidad o que no se va a utilizar algún sensor se ha utilizado conectores servo tipo JR.



Figura 45: Imagen conector servo tipo JR [Fuente: <http://speedrcmaresme.es>]



Capítulo 8

Diseño Mecánico

8.1 Introducción

8.2 Skybot

8.3 Diseño del robot

8.4 Proceso de fabricación

8.1 INTRODUCCIÓN

Para diseñar el diseño de este robot se ha basado en el Skybot y en otro proyecto de fin de carrera el cual consistía en una evolución del robot skybot añadiéndole nuevos sensores y haciendo las piezas imprimibles.

A rasgos generales se ha modificado prácticamente todo el diseño ya que se ha incluido algún sensor más y la electrónica es como se ha dicho totalmente diferente ya que se emplea la placa arduino.

La novedad más grande que se ha realizado es que se ha separado en dos partes el chasis y la base de los sensores con el fin de que si en un futuro se quieren incorporar otros sensores solo se tendría que cambiar la base donde van situados manteniendo el chasis.

Se va a describir como es el Skybot a rasgos generales para que se vean las diferencias y novedades que se ha incluido, como la descripción de las modificaciones que se ha hecho, en definitiva se describirá el diseño del robot.

8.2 SKYBOT

La estructura mecánica es fácilmente replicable y se pueden emplear diversos materiales para su fabricación como madera, aluminio etc. Esta estructura está compuesta por 7 piezas de metacrilato de 3mm de grosor, dos servos Futaba 3003 modificados para girar 360º y una rueda loca. Sobre la cual están montadas una placa Skypic y una placa de alimentación SKY293. Las dimensiones son 16X11X10cm.

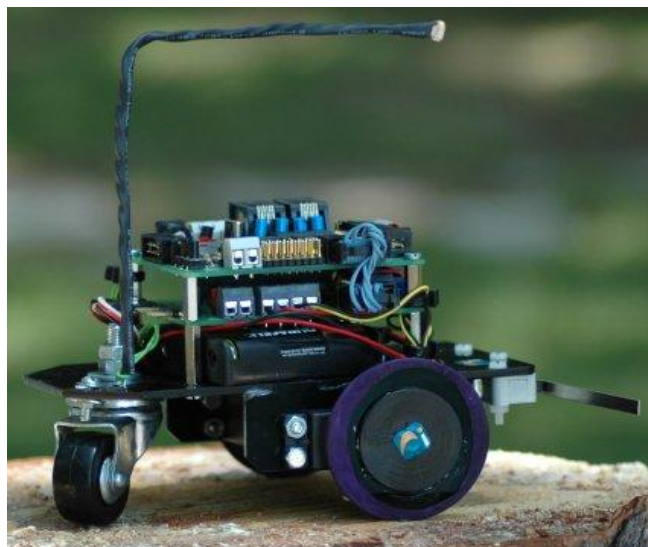


Figura 46: Imagen de un Skybot [Fuente: <http://www.learobotics.com>]

8.3 DISEÑO DEL ROBOT

8.3.1. Base sensores:

Como ya se ha dicho se ha diseñado una base para situar los sensores, independiente de los motores y la rueda loca. Las dimensiones de la base son 148X90X3mm y se ha impreso en dos partes la primera tiene 70X90x3mm y la segunda 78X90X3mm. A continuación se muestra como es el diseño de las dos partes de la base sensores (Son imágenes sacadas de openscad).

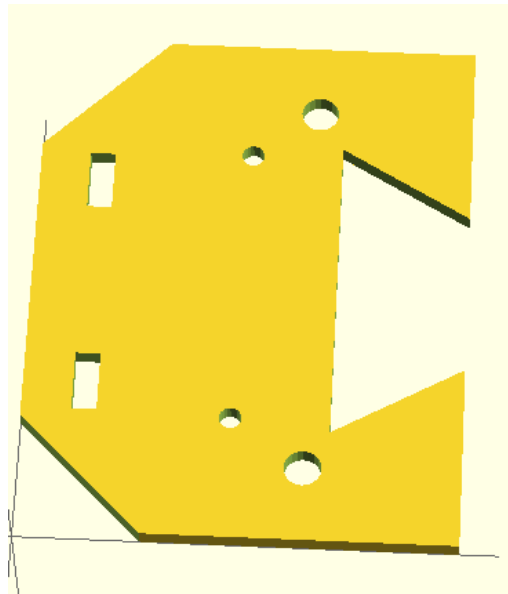


Figura 47: Parte trasera de la base sensores [Fuente: Elaboración propia]

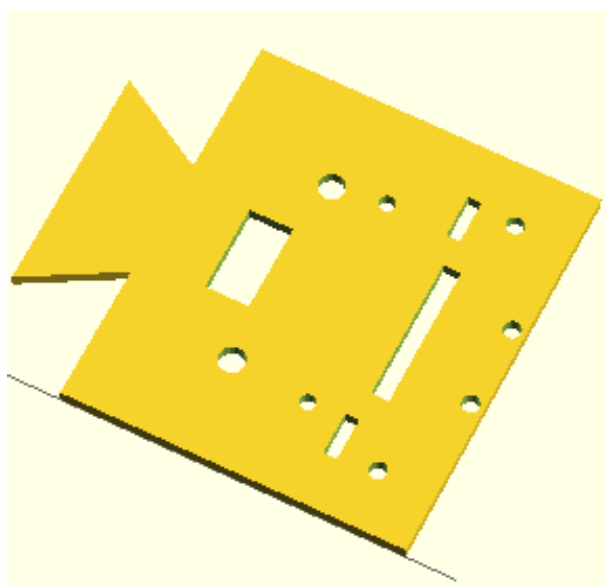


Figura 48: Parte delantera de la base sensores [Fuente: Elaboración propia]

8.3.2. Base para el sensor Sharp y base del sensor de ultrasonidos

Para diseñar estas piezas se ha basado en las dimensiones del sensor de ultrasonido SFR05 y en los taladros del sensor de distancia Sharp donde atornillemos dicho sensor. Las dimensiones de la base del sensor ultrasonido son 50X30X3mm y las del sensor Sharp son 10X17X3mm.

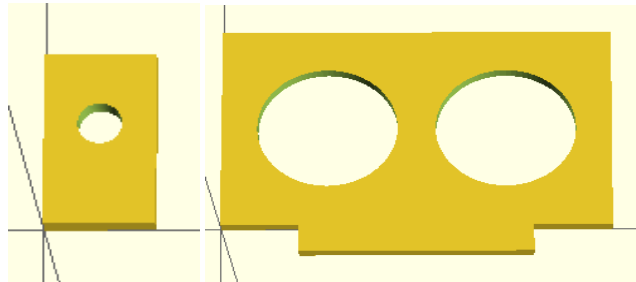


Figura 49: Imagen de la Base para el sensor Sharp (izquierda) y base del sensor de ultrasonidos (derecha) [Fuente: Elaboración propia]

8.3.3. Base del sensor CNY70

Partiendo de las dimensiones del sensor se ha creado esta pieza con dos raíles para ajustar más abajo o arriba los sensores ya que estos deben de estar lo más próximo al suelo para identificar mejor lo negro de lo blanco y otro carril para pasar por ahí los conectores que irán a la placa impresa.

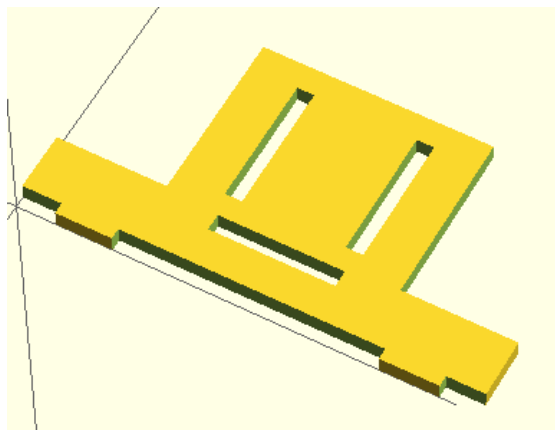


Figura 50: Imagen de la Base del sensor CNY70 [Fuente: Elaboración propia]

8.3.4. Ruedas

Las ruedas que hemos empleado no las hemos diseñado nosotros ya que iban a ser muy similares o iguales, por ello hemos reutilizado unas ya

impresas. Tienen un diámetro de 55mm. Tardaron en imprimirse las dos sobre hora y media.



Figura 51: Imagen de la rueda del robot [Fuente: Elaboración propia]

8.3.5. Chasis del robot:

En él se incorpora la rueda loca que en lugar de ser una rueda clásica se emplea una canica. Para ello se ha diseñado un soporte como se puede ver en la imagen que se muestra a continuación. También tiene los taladros donde irán situados los motores con las ruedas. Las dimensiones del chasis son 90X90X30mm, el grosor de las paredes del chasis es de 5mm mientras que la base es de 3mm. La canica que se emplea es un rodamiento de acero con un diámetro de 11mm.

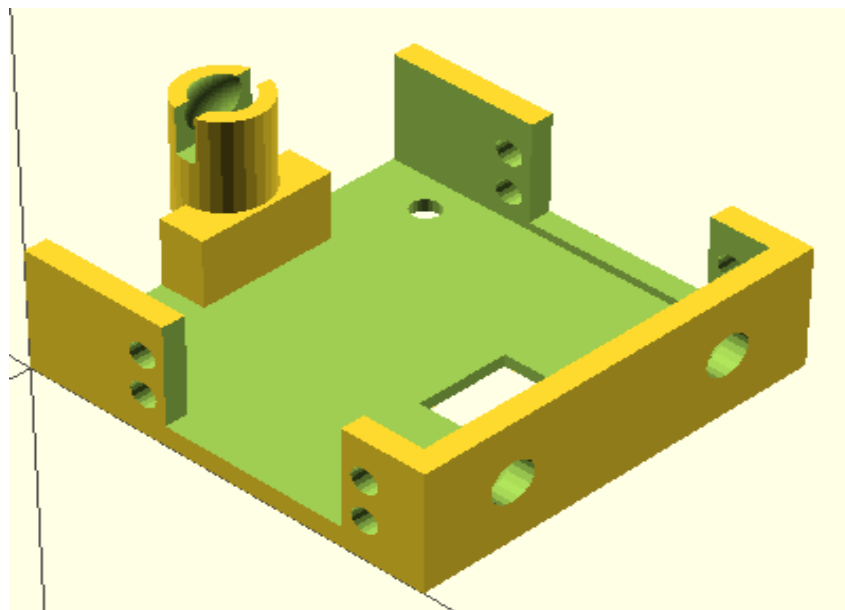


Figura 52: Chasis del robot [Fuente: Elaboración propia]

8.4 PROCESO DE FABRICACIÓN

Una vez que se ha diseñado las piezas en este caso desde el software openscad se exporta el diseño en .stl para que desde el programa slic3r lo exportemos a gcode que será el que se le pase a pronterface.

A la impresora 3D se la pasa desde el programa pronterface las instrucciones para fabricar cada pieza. También se indicara a la temperatura a la que debe de calentarse el extrusor y la base, en este caso se calentaran a 220°C y 120°C respectivamente.

Para todas las piezas se ha empleado los mismos parámetros de impresión ya que se le ha dado a todas la misma rigidez a pesar de que a las bases de los sensores hubiese servido menos solidez. A continuación se describirán algunos de los parámetros más importantes que se ha utilizado y posteriormente se indicara en una tabla el valor de esos parámetros que se han puesto.

- **Infill (mm/s):**

Velocidad para el relleno. El relleno puede ir a una velocidad mayor a la del perímetro. Se pueden utilizar velocidades de hasta 60 mm/s pero si el filamento se corta o no se deposita bien hay que reducirla.

- **Perimeters:**

Número de líneas en el perímetro. Con 3 líneas se consiguen piezas fuertes aunque en ocasiones para piezas con columnas muy finas hace que la impresión salga mal, por lo que en estos casos hay que disminuir el valor a 2 o 1.

- **Solid layers:**

Numero de capas con relleno 100% al principio y al final de una pieza.

- **Fill density:**

Porcentaje de relleno en el resto de capas. Normalmente se usa 0.4 que significa relleno al 40%. Si se quiere conseguir piezas más fuertes hay que aumentar este valor, el inconveniente es que se aumenta el tiempo de impresión o al revés si se quiere piezas más ligeras se pueden usar valores inferiores.

- **Layer height (mm):**

Altura de cada capa. Este valor afecta mucho al nivel de detalle que tendrá la pieza impresa es decir la precisión. Un valor adecuado sobretodo en las primeras impresiones es de 0.4 cuando se usa un hotend (Es un dispositivo

encargado de empujar un hilo de plástico una vez fundido, ya que este se calienta hasta fundir el plástico) de 0.5mm de salida.

Para conseguir más detalle hay que reducir este valor a 0.35. 0.3 o incluso menos.

- **Generate support material:**

Para imprimir piezas con partes “volantes” es necesario imprimir antes un soporte. Slic3r calcula los soportes necesarios si se activa esta opción.

Tabla con los valores que se ha usado para imprimir las piezas:

Parámetro	Valor
Infill (mm/s)	40
Perimeters	3
Solid layers	3
Fill density	0.4
Layer height	0.35
Generate support material	Desactivado

Tabla 1: Valores de Impresión 3D

El software Slic3r permite generar el Gcode de dos piezas a la vez para que se impriman juntas. En este caso esta función se ha utilizado para el soporte del sensor Sharp ya que era una pieza muy pequeña y se podían imprimir a la vez. A continuación se muestra la imagen.

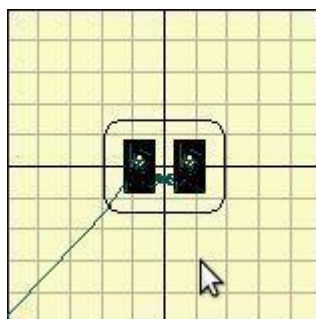


Figura 53: Imagen de impresión doble [Fuente: Elaboración propia]

La siguiente Tabla muestra el tiempo de impresión de cada pieza de un robot:

PIEZA	TIEMPO IMPRESIÓN(Min)
Parte 1 de la base sensores	46:00
Parte 2 de la base sensores	55:45
Base inferior	3:00:00
Bases sensor sharp	10:00
Base sensor ultrasonido	15:45
Base sensor CNY70	23:00
Total= 4:50:40	

Tabla 2: Tiempo de impresión de un robot

El resultado de las piezas definitivas ya que ha habido piezas que las hemos tenido que desechar por que no salían bien.

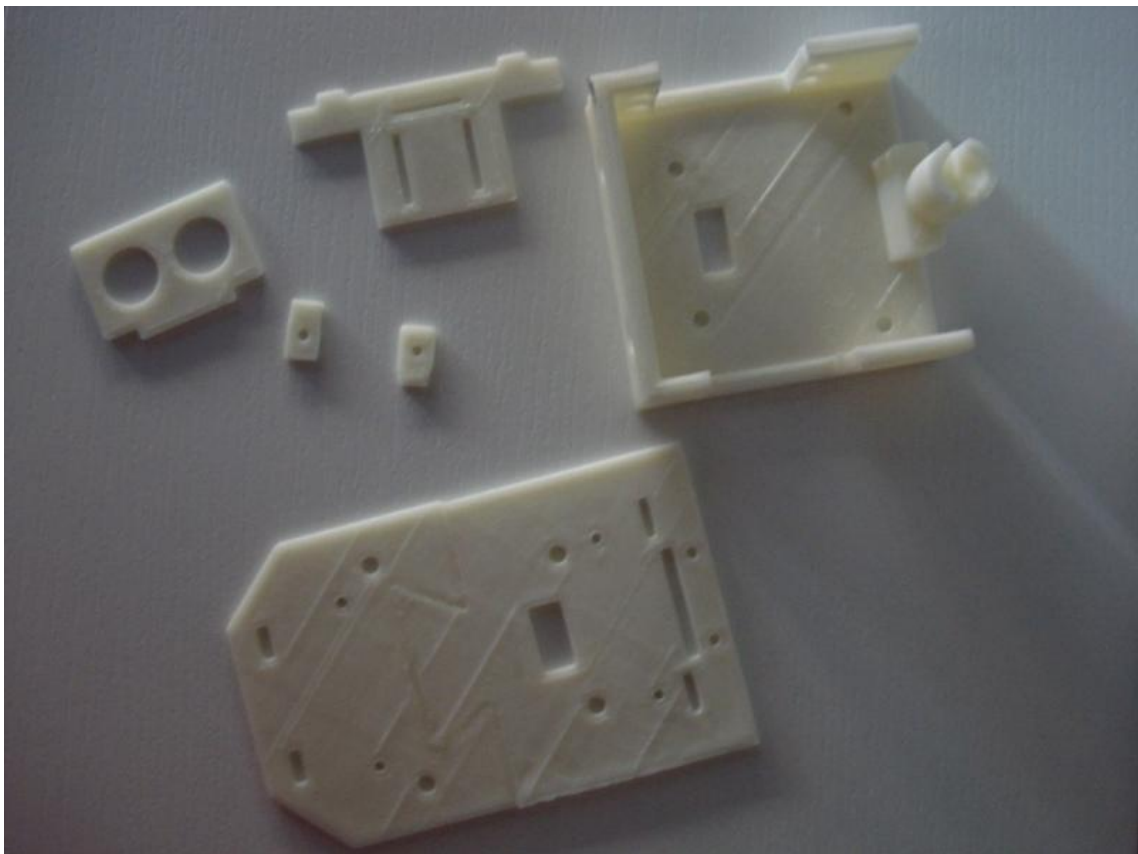


Figura 54: Imagen piezas del robot impresas [Fuente: Elaboración propia]

Capítulo 9

Programación

9.1 Introducción

9.2 Calibración de sensores

9.3 Función de los sensores según la programación

9.4 Posibles tareas que puede realizar el robot

9.1 INTRODUCCIÓN

El objetivo es hacer un programa para que el robot siga líneas negras es decir un sigue líneas y utilizar los sensores descritos en el capítulo 4 para que el robot efectué determinados cambios ante algún obstáculo o ante un cambio de iluminación.

Para llevar a cabo este objetivo primeramente se han analizado los sensores que se iban a utilizar, el acondicionamiento necesario para conectarlos a la placa Arduino y la forma en la que había que programar cada sensor. También si era necesario o no calibrar alguno previamente.

En este capítulo se mostrara la función que realiza cada sensor según se ha programado y cuál de los sensores se ha tenido que calibrar.

9.2 CALIBRACIÓN DE SENSORES

Es necesario calibrar el sensor Sharp ya que según a la distancia a la que se encuentre un obstáculo o una pared nos devuelve un valor u otro siempre comprendidos entre 0 y 1023. Por eso se ha realizado un programa y se ha obtenido los valores a determinadas distancias. Las cuales se muestran en la siguiente tabla.

Distancia Obstáculo (mm)	Valor del sensor
100	410
200	550
300	438
400	320
500	265

Tabla 3: Calibración del sensor Sharp

Como en el caso del sensor anterior también es necesario calibrar las LDR que cumplen una función importante que se explicara posteriormente. Estas según cambia la iluminación nos devuelve un valor entre 0 y 1023. En la siguiente tabla se muestran los valores que se han obtenido según la intensidad de la luz.

Intensidad de la luz	Valor del sensor
Luz Intensa	500
Luz Eléctrica (Bombilla)	910
Luz Normal (Calle)	925
Media Luz	999
Luz Apagada	1023

Tabla 4: Calibración del sensor de luz (LDR)

9.3 FUNCIÓN DE LOS SENSORES SEGÚN LA PROGRAMACIÓN

9.3.1 Sensor de contacto y Fotorresistencia:

La función del Bumper es que si el robot choca contra algún objeto o pared este se pare y continúe avanzando pudiendo ocasionar desperfectos en el robot.

En cuanto a la misión de las fotorresistencias o LDR es que como los sensores encargados de detectar la línea negra, es decir los CNY70, son muy sensibles a la iluminación se ha programado de tal forma que si la intensidad de luz disminuye o aumenta demasiado el robot se parara hasta que la iluminación sea la correcta.

9.3.2 Sensor CNY70 y Sensor Sharp:

La función de los sensores CNY70 es detectar las líneas negras o blancas y según se detecte negro o no las ruedas cambiarán de dirección para corregir la trayectoria.

El sensor Sharp detectará la distancia a la que se encuentre de cualquier obstáculo según sea esta distancia el robot se parará para no chocar con ello.

9.3.2 Sensor de ultrasonidos:

La programación de este sensor es más especial. Primeramente recordar que se ha utilizado el modo 2 explicado ya en el capítulo 4. En este modo la señal de operación y la señal del eco se envían por la misma terminal, por lo tanto es necesario configurar el pin como emisor y receptor. Para poder medir la distancia en el Arduino se utiliza una función llamada "Pulse in" la cual lee un pulso en estado HIGH o LOW.

Es necesario tener en cuenta que el eco que se recibe es un pulso por lo tanto su ancho es proporcional a la distancia con respecto al objeto, este sensor si no detecta nada en 30 μ s su nivel cambia a estado bajo.

Para calcular la distancia en cm se debe tener en cuenta que la velocidad del sonido es 340m/s o 29 μ s/cm. El sonido llega hasta el objeto, rebota y regresa como eco por lo que se toma el tiempo en milisegundos y se divide entre 29 para obtener la distancia y después se divide entre 2 ya que el ping viaja ida y vuelta.

Si se quiere en lugar de cm calcular la distancia en pulgadas es necesario saber que el sonido viaja a 1130 pies por segundo y se recorren 73746 microsegundos por pulgada, por lo que es necesario dividir la señal que nos da entre 74 para obtener la distancia total y finalmente dividir entre 2 para obtener la distancia del sensor al objeto.

En el código del programa hay que aplicar las siguientes formulas explicadas anteriormente:

```
long Pulgadas (long microsegundos)
{
    return microsegundos/74/2;
}

long centimetros (long microsegundos)
{
    return microsegundos/29/2;
}
```

Finalmente decir que la función de este sensor es similar a la del sensor Sharp, es decir según la distancia a la que se encuentre el objeto el robot se para o retrocederá hasta poder continuar su trayectoria.

9.4 POSIBLES TAREAS QUE PUEDE REALIZAR EL ROBOT

Según la programación que se realice el micro-robot podrá hacer una cosa u otra, ya que con los sensores que se emplean se puede programar el robot para distintas aplicaciones. A continuación se va a exponer algunas de las tareas que podría realizar el robot.



- La tarea que se ha programado en este caso ha sido la del seguir líneas, es decir se marca un recorrido y este tiene que seguirlo. Los sensores de luz se emplean por si esta baja demasiado o aumenta mucho para que el robot se pare. También se utilizan el sensor de ultrasonido y el sensor Sharp para detectar obstáculos y así evitar que el robot choque con ellos.
- Otra tarea posible seria mediante las LDR que el robot buscase la luz y se aproximase a ella.
- También podemos reprogramar el sensor ultrasonido y el sensor Sharp y hacer que el robot se aproxime o se aleje de los objetos que se va encontrando a su paso.

Decir que como se puede ver el robot según se empleen unos sensores u otros se pueden realizar bastantes cosas con él.

Capítulo 10

Planificación y Presupuesto

10.1 Planificación

10.2 Costes de ejecución

10.3 Importe total del presupuesto

En este capítulo se va a mostrar la planificación mediante el diagrama de Gantt y el presupuesto desglosado del proyecto, especificando los diferentes gastos que han sido necesarios para su realización.

10.1 PLANIFICACIÓN

10.1.1 Resumen de horas dedicadas y diagrama de Gantt

Para realizar la planificación se ha utilizado una herramienta básica en la rama de ingeniería del software; el diagrama de Gantt. Este es una herramienta grafica utilizada para mostrar el tiempo previsto de esfuerzo para realizar diferentes tareas o actividades a lo largo de un tiempo total determinado.

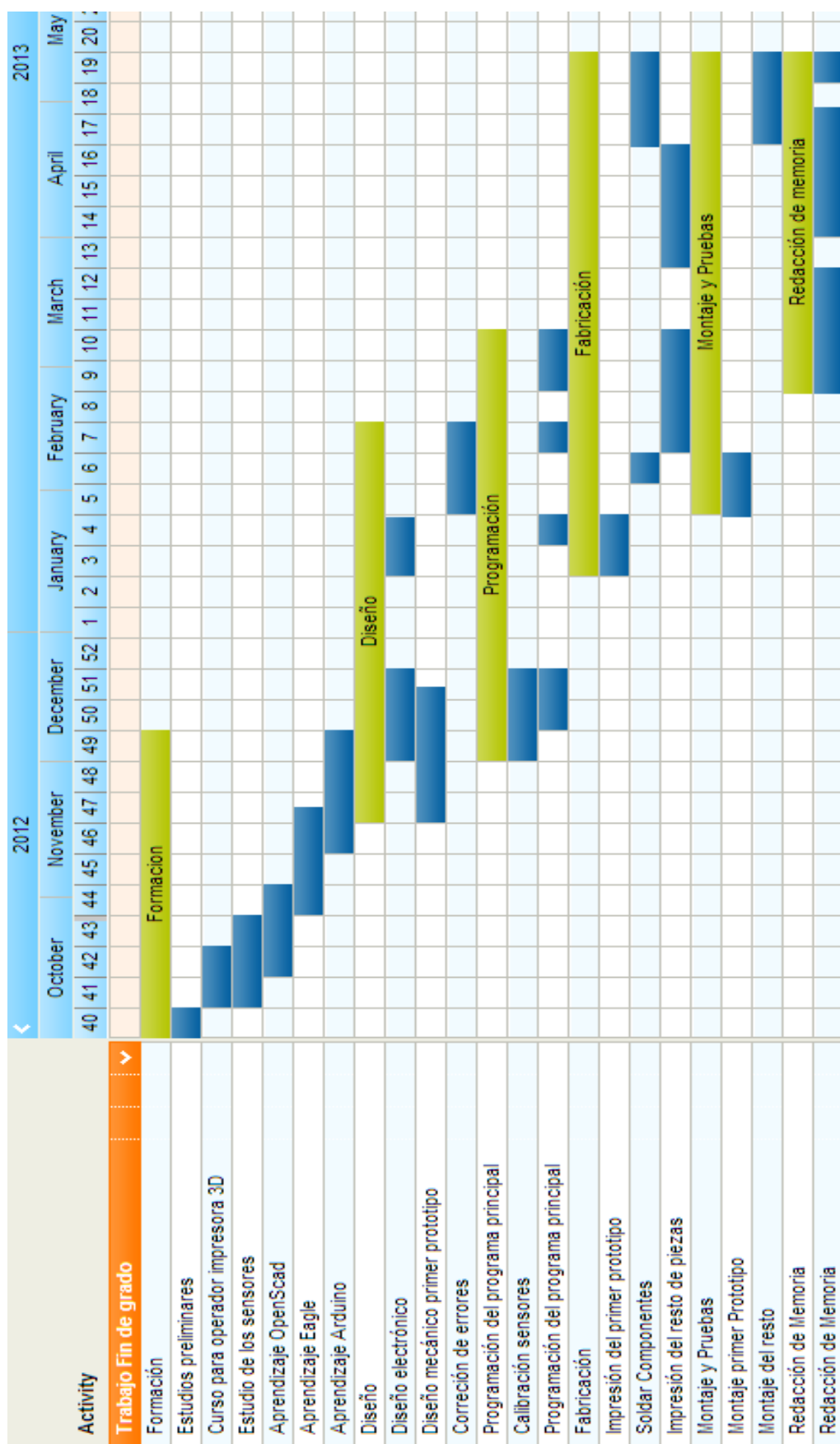
A continuación se expone el número aproximado de horas dedicadas al trabajo.

- **Fase de formación:** 10 semanas X 13 horas = 130 horas
- **Fase de diseño (electrónico y mecánico):** 10 semanas X 15 horas=150 h
- **Fase de programación:** 7 semanas X 5 horas = 35 horas
- **Fase de fabricación:** 14 semanas X 7 horas = 98 horas
- **Fase de montaje y pruebas:** 5 semanas X 6.5 horas = 32,5 horas
- **Fase de redacción:** 9 semanas X 14 horas = 126 horas

Por tanto el número de horas totales dedicadas al trabajo será la suma de las horas dedicadas a cada una de las fases. El coste aproximado en horas de la totalidad del trabajo es de 572 horas.

Una vez que se ha hecho un resumen de las horas dedicadas al trabajo, se indica mediante el diagrama de Gantt la planificación de esas semanas más detalladamente.

Diagrama de Gantt:



10.2 COSTES DE EJECUCIÓN

Estos costes incluyen todos los costes asociados a material de laboratorio, software y tiempo empleado para el desarrollo del proyecto.

10.2.1 Costes por material de laboratorio

Equipo	Cantidad	Precio Ud. €	Periodo de Amortización (Años)	Periodo de uso (Años)	Precio Total €
Ordenador	1	600€	5	0.8	96€
Multímetro digital Salki	1	14€	5	0.7	1.96€
Soldador	1	10€	2	0.5	2.50€
Impresora 3D	1	450€	5	0.6	54€
					Total=154.46€

Tabla 5: Costes por material de laboratorio

10.2.2 Costes por software

Software	Cantidad	Precio Ud. €	Periodo de Amortización (Años)	Periodo de uso (Años)	Precio Total €
OpenScad	1	0€	-	-	0€
Arduino	1	0€	-	-	0€
Eagle	1	0€	-	-	0€
Slic3r y Pronterface	1	0€	-	-	0€
Office 2013	1	100€	2	0.7	35€
					Total=35€

Tabla 6: Costes por software

10.2.3 Costes por hardware

Hardware	Cantidad	Precio Ud. €	Precio Total €
Placas de circuito Impreso	10	24,80€	248€
Placa Arduino mega 2560	1	41,00€	41,00€
Componentes Electrónicos	-	-	66.00€
Sensores Ultrasonido SRF05	2	14.80€	29.60€
Sensores de luz LDR	4	0.80€	3.20€
Sensores de contacto Bumper	4	0.40€	1.60€
Sensores Infrarrojos CNY70	4	0.88€	3.50€
Servos Futaba 3003	4	4,00€	16.00€
Sensores Sharp	2	16.90€	33.80€
Plástico ABS	2Kg	30,00€	30,00€
			Total=472.7€

Tabla 7: Costes por hardware

10.2.4 Costes por personal

Función	Nº Horas	Precio Ud. €	Precio Total €
Ingeniería	200	20€	4000
			Total=4000€

Tabla 8: Costes por personal

10.2.5 Coste total de presupuesto de ejecución

Costes	Precio Total €
Costes por material de laboratorio	154.46€
Costes por software	35€
Costes por hardware	472.7€
Costes por personal	4000€
	Total=4662€

Tabla 9: Coste total de presupuesto de ejecución



10.3 IMPORTE TOTAL DEL PRESUPUESTO

Concepto	Precio Total €
Coste total de presupuesto de ejecución	4662€
Honorarios por redacción	327€
	Total=4989€

Tabla 10: Importe total del presupuesto

Siendo los honorarios de redacción el 7% del presupuesto de ejecución.

“El presupuesto total de este proyecto asciende a la cantidad de:

Cuatro mil novecientos ochenta y nueve”

Leganés a 11 de Mayo de 2013

El ingeniero proyectista

Fdo. Ángel Mate Martínez

Conclusiones y trabajos futuros

Tras la realización de este trabajo de fin de grado se pueden extraer las siguientes conclusiones:

- Ha sido un trabajo de duro aprendizaje durante meses, pero ha merecido la pena porque ahora he adquirido muchos más conocimientos del mundo de la robótica. Además la realización de este trabajo me ha permitido conocer el mundo de Arduino y las impresoras 3D que me llamaba la atención. También he tenido la oportunidad de diseñar una placa impresa que yo por la elección de las asignaturas optativas no había realizado.
- Decir que tras imprimir las piezas se puede afirmar que tienen una buena calidad. La única pega que se podría poner es que al usar principalmente una impresora 3D casera se necesitaba a menudo de mantenimiento y eso retrasaba la impresión.
- Destacar los contenidos open source que se comparten abiertamente y ayudan a mejorar las cosas de una manera más sencilla y solucionar problemas más fácilmente ya que se pueden observar cosas resueltas.
- Por último señalar que todos los objetivos marcados han sido superados.

De cara al trabajo futuro:

- Se podría incorporar algún sensor más o reubicarlo para realizar otras tareas. En el caso de querer reubicar algún sensor solo sería necesario rediseñar la base de los sensores ya que el chasis es independiente. Como se ha expuesto es una de las ventajas de este micro-robot.
- También se podría rediseñar el robot para minimizar su tamaño y así conseguir un menor tiempo de fabricación.

Capítulo 11

Bibliografía y Referencias

11.1 Referencias relacionadas con sensores y motores

11.2 Referencias relacionadas con la impresión 3D

11.3 Referencias relacionadas con el software

11.1 Referencias relacionadas con los sensores y motores

- <http://www.ardumania.es/midiendo-distancias-con-un-sensor-de-ultrasonidos/> (sensor de ultrasonido) (30/04/2013)
- http://www.jmnlab.com/cny70/cny_70.html (sensor CNY70) (30/04/2013)
- <http://arduino.cc/forum/index.php/topic,32072.0.html> (sensor CNY70) (07/05/2013)
- http://arduino.cc/es_old/Ejercicios/LDR (sensor LDR) (30/04/2013)
- <https://sites.google.com/site/2007arduino/example-code/sensor-tutorials/switch-bump-sensor> (sensor Bumper) (07/05/2013)
- http://arduino.cc/es_old/Pduino/EJ2 (sensor Sharp) (07/05/2013)
- <http://www.slideshare.net/ilusaud/minibloq-arduino> (Tutorial donde se explica el conexionado y la programación de motores y algunos sensores) (30/04/2013)
- <http://www.learobotics.com/proyectos/cuadernos/ct2/ct2.html> (Trucar servos futaba 3003) (30/04/2013)
- <http://garagelab.com/profiles/blogs/tutorial-l293d-h-bridge-dc-motor-controller-with-arduino> (Controlar motores con driver L293D) (07/05/2013)
- http://arduino.cc/es_old/L293D/L293D (Control motores puente H) (30/04/2013)

11.2 Referencias relacionadas con la Impresión 3D

- http://es.wikipedia.org/wiki/Impresi%C3%B3n_3D (Información sobre la impresión 3D, historia y evolución) (17/04/2013)
- <http://learobotics.com/alberto/lib/exe/fetch.php?media=theses:arturovera.pdf> (Información sobre la impresora 3D) (17/04/2013)
- http://www.reprap.org/wiki/Clone_Wars:_Guia_de_calibracion/es (Calibración de la impresora 3D) (17/04/2013)
- http://reprap.org/wiki/Prusa_Air/es (Información sobre modelo impresora 3D) (17/04/2013)



- http://es.wikipedia.org/wiki/Proyecto_RepRap (30/04/2013)
- <http://reprapworld.com/> (Sitio web donde se puede comprar componentes para impresora 3D y el plástico ABS) (30/04/2013)
- <http://www.thingiverse.com/> (Portal sobre impresión 3D con mucha información e ideas sobre diseño) (30/04/2013)
- <http://asrob.uc3m.es> (Portal de la asociación de robótica de la uc3m con información sobre la impresión 3D) (30/04/2013)
- <http://www.learobotics.com> (Portal de robótica con toda la información sobre el Skybot entre otros) (30/04/2013)

11.3 Referencias relacionadas con el software

- <http://www.tucamon.es/contenido/avanzando-5> (Tutoriales para programar Arduino) (07/05/2013)
- <http://www.todopic.com.ar/foros/index.php?topic=24080.0> (Crear archivos GERBER desde Eagle) (07/05/2013)
- http://en.wikibooks.org/wiki/OpenSCAD_User_Manual (Manual de Openscad) (07/05/2013)
- http://www.datsi.fi.upm.es/docencia/Micro_C/eagle/tutorial-spa.pdf (Tutorial Eagle) (07/05/2013)
- <http://slic3r.org/> (Página oficial slic3r) (07/05/2013)
- <http://www.tomsplanner.com/> (Pagina para realizar diagrama de Gantt) (07/05/2013)

Capítulo 12

Anexos

12.1 Hojas de características del sensor CNY70

12.2 Hojas de características del sensor Sharp

12.3 Hojas de características del driver L293

12.4 Esquemático de la shield

12.1 Hoja de características del sensor CNY70



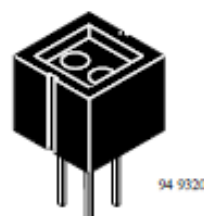
CNY70

Vishay Semiconductors

Reflective Optical Sensor with Transistor Output

Description

The CNY70 has a compact construction where the emitting light source and the detector are arranged in the same direction to sense the presence of an object by using the reflective IR beam from the object. The operating wavelength is 950 nm. The detector consists of a phototransistor.

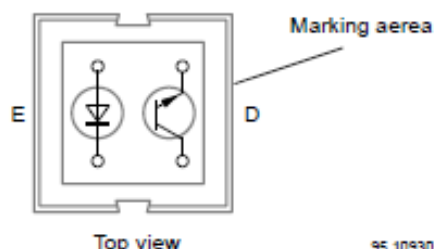


Applications

- Optoelectronic scanning and switching devices i.e., index sensing, coded disk scanning etc. (optoelectronic encoder assemblies for transmission sensing).

Features

- Compact construction in center-to-center spacing of 0.1"
- No setting required
- High signal output
- Low temperature coefficient
- Detector provided with optical filter
- Current Transfer Ratio (CTR) of typical 5%



Order Instruction

Ordering Code	Sensing Distance	Remarks
CNY70	0.3 mm	

CNY70

Vishay Semiconductors



Absolute Maximum Ratings

Input (Emitter)

Parameter	Test Conditions	Symbol	Value	Unit
Reverse voltage		V_R	5	V
Forward current		I_F	50	mA
Forward surge current	$t_p \leq 10 \mu s$	I_{FSM}	3	A
Power dissipation	$T_{amb} \leq 25^\circ C$	P_V	100	mW
Junction temperature		T_j	100	$^\circ C$

Output (Detector)

Parameter	Test Conditions	Symbol	Value	Unit
Collector emitter voltage		V_{CEO}	32	V
Emitter collector voltage		V_{ECO}	7	V
Collector current		I_C	50	mA
Power dissipation	$T_{amb} \leq 25^\circ C$	P_V	100	mW
Junction temperature		T_j	100	$^\circ C$

Coupler

Parameter	Test Conditions	Symbol	Value	Unit
Total power dissipation	$T_{amb} \leq 25^\circ C$	P_{tot}	200	mW
Ambient temperature range		T_{amb}	-55 to +85	$^\circ C$
Storage temperature range		T_{stg}	-55 to +100	$^\circ C$
Soldering temperature	2 mm from case, $t \leq 5 s$	T_{sd}	260	$^\circ C$



CNY70

Vishay Semiconductors

Electrical Characteristics ($T_{amb} = 25^{\circ}\text{C}$)

Input (Emitter)

Parameter	Test Conditions	Symbol	Min.	Typ.	Max.	Unit
Forward voltage	$I_F = 50 \text{ mA}$	V_F		1.25	1.6	V

Output (Detector)

Parameter	Test Conditions	Symbol	Min.	Typ.	Max.	Unit
Collector emitter voltage	$I_C = 1 \text{ mA}$	V_{CEO}	32			V
Emitter collector voltage	$I_E = 100 \mu\text{A}$	V_{ECO}	5			V
Collector dark current	$V_{CE} = 20 \text{ V}, I_F = 0, E = 0$	I_{CEO}			200	nA

Coupler

Parameter	Test Conditions	Symbol	Min.	Typ.	Max.	Unit
Collector current	$V_{CE} = 5 \text{ V}, I_F = 20 \text{ mA}, d = 0.3 \text{ mm}$ (figure 1)	$I_C^{(1)}$	0.3	1.0		mA
Cross talk current	$V_{CE} = 5 \text{ V}, I_F = 20 \text{ mA}$ (figure 1)	$I_{CX}^{(2)}$			600	nA
Collector emitter saturation voltage	$I_F = 20 \text{ mA}, I_C = 0.1 \text{ mA}, d = 0.3 \text{ mm}$ (figure 1)	$V_{CEsat}^{(1)}$			0.3	V

¹⁾ Measured with the "Kodak neutral test card", white side with 90% diffuse reflectance

²⁾ Measured without reflecting medium

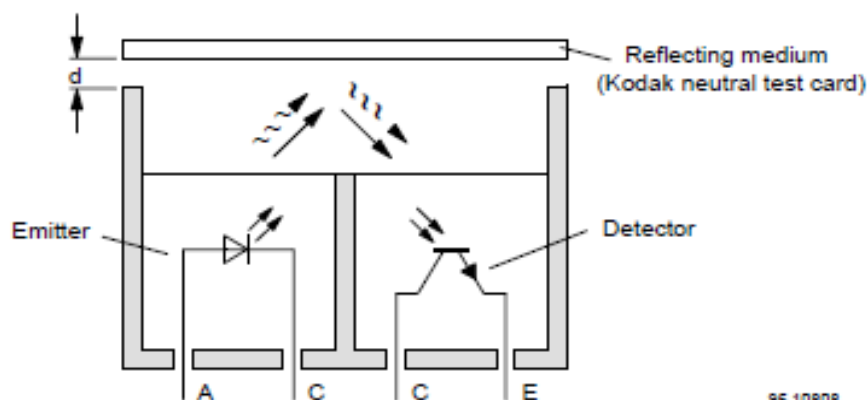


Figure 1. Test circuit

95 10808

CNY70

Vishay Semiconductors



Typical Characteristics ($T_{amb} = 25^{\circ}\text{C}$, unless otherwise specified)

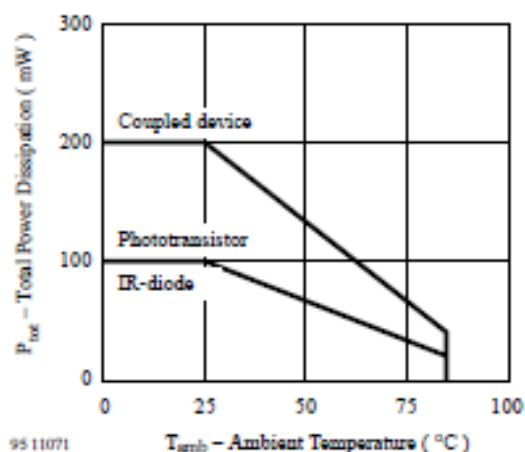


Figure 2. Total Power Dissipation vs. Ambient Temperature

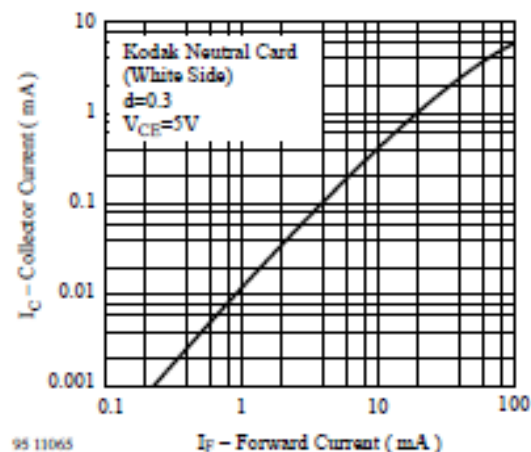


Figure 5. Collector Current vs. Forward Current

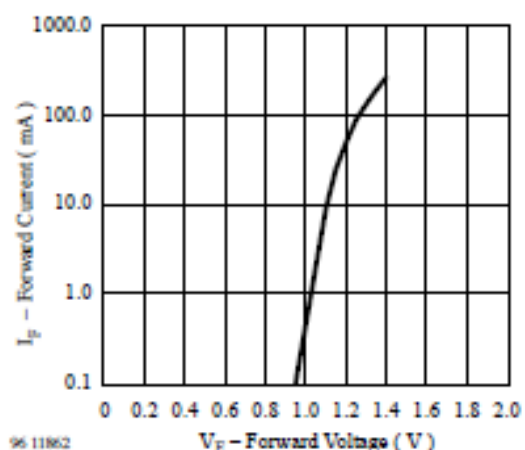


Figure 3. Forward Current vs. Forward Voltage

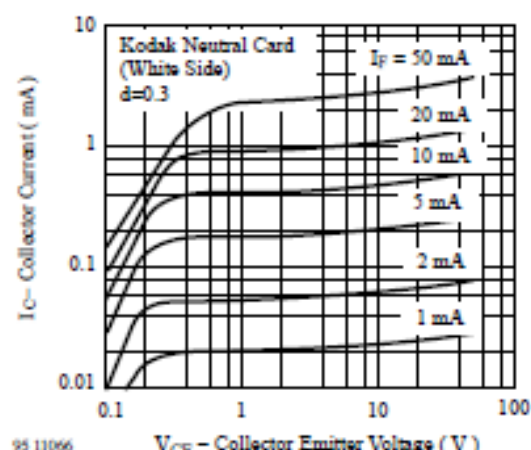


Figure 6. Collector Current vs. Collector Emitter Voltage

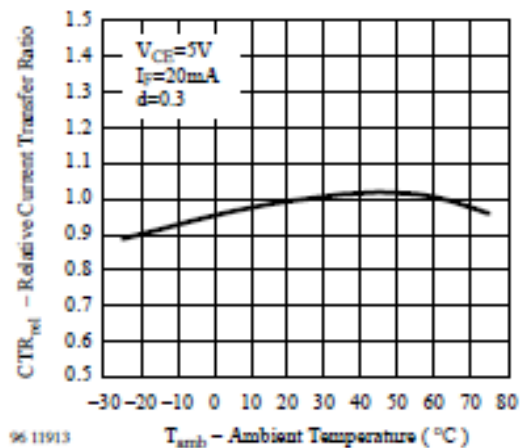


Figure 4. Relative Current Transfer Ratio vs. Ambient Temperature

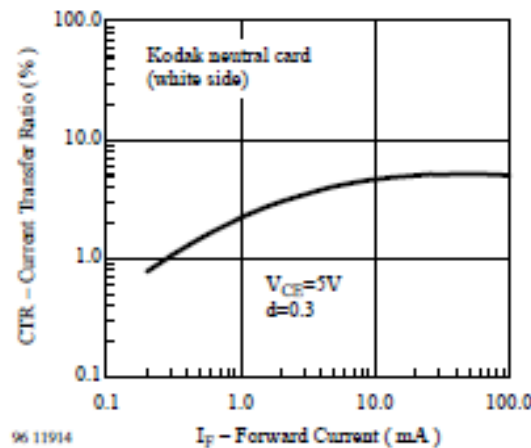


Figure 7. Current Transfer Ratio vs. Forward Current



CNY70

Vishay Semiconductors

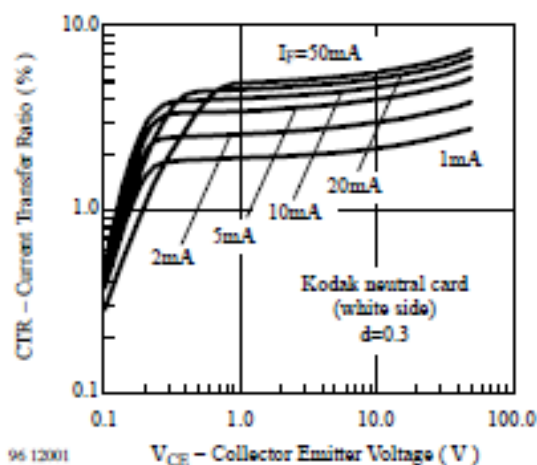


Figure 8. Current Transfer Ratio vs. Collector Emitter Voltage

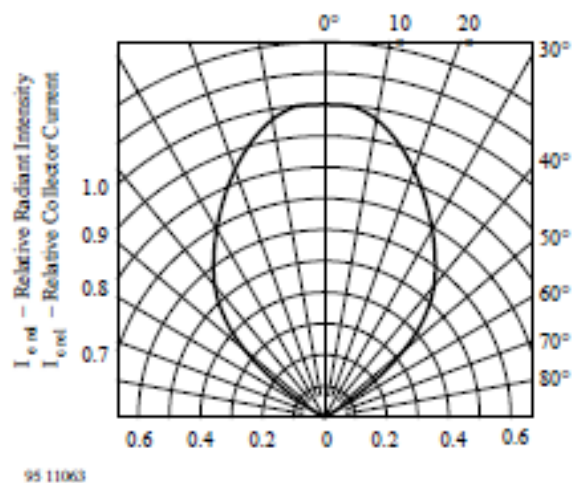


Figure 10. Relative Radiant Intensity/Collector Current vs. Displacement

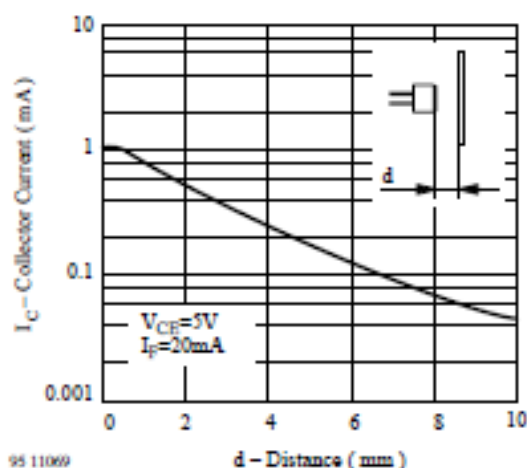


Figure 9. Collector Current vs. Distance

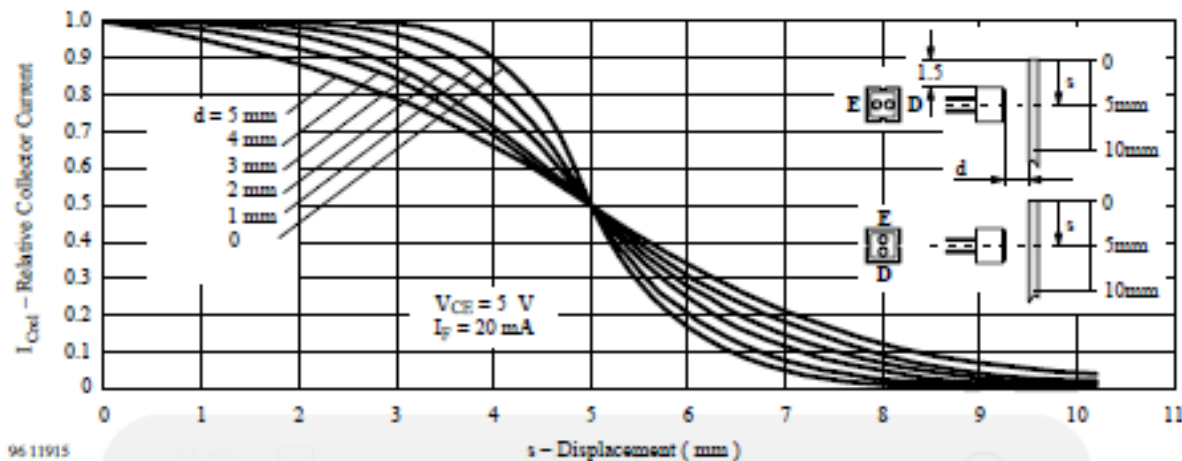


Figure 11. Relative Collector Current vs. Displacement

12.2 Hoja de características del sensor Sharp

SHARP

GP2Y0A02YK0F

GP2Y0A02YK0F

Distance Measuring Sensor Unit
Measuring distance: 20 to 150 cm
Analog output type



■Description

GP2Y0A02YK0F is a distance measuring sensor unit, composed of an integrated combination of PSD (position sensitive detector), IRED (infrared emitting diode) and signal processing circuit.

The variety of the reflectivity of the object, the environmental temperature and the operating duration are not influenced easily to the distance detection because of adopting the triangulation method.

This device outputs the voltage corresponding to the detection distance. So this sensor can also be used as a proximity sensor.

■Features

1. Distance measuring range : 20 to 150 cm
2. Analog output type
3. Package size : 29.5×13×21.6 mm
4. Consumption current : Typ. 33 mA
5. Supply voltage : 4.5 to 5.5 V

■Agency approvals/Compliance

1. Compliant with RoHS directive (2002/95/EC)

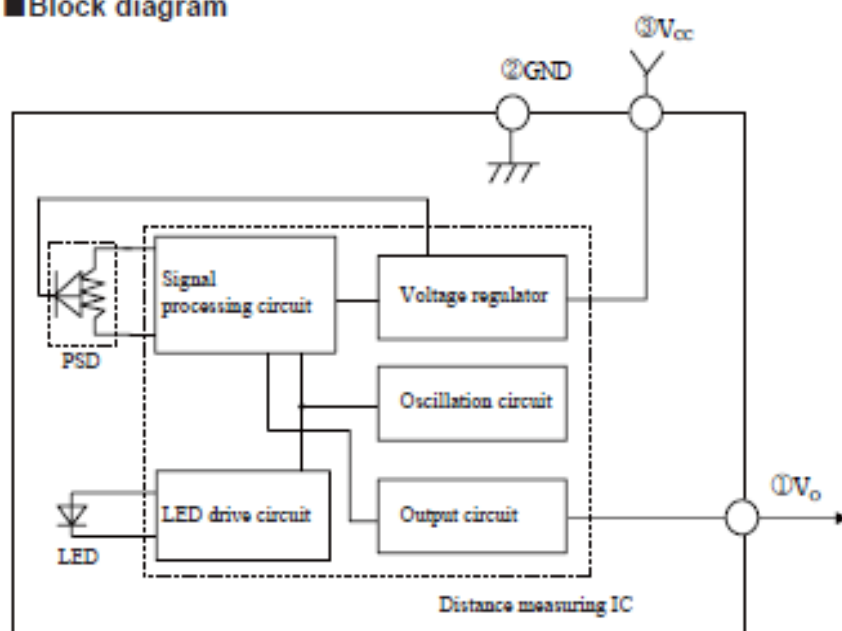
■Applications

1. Touch-less switch
(Sanitary equipment, Control of illumination, etc.)
2. Sensor for energy saving
(ATM, Copier, Vending machine, Laptop computer, LCD monitor)
3. Amusement equipment
(Robot, Arcade game machine)

SHARP

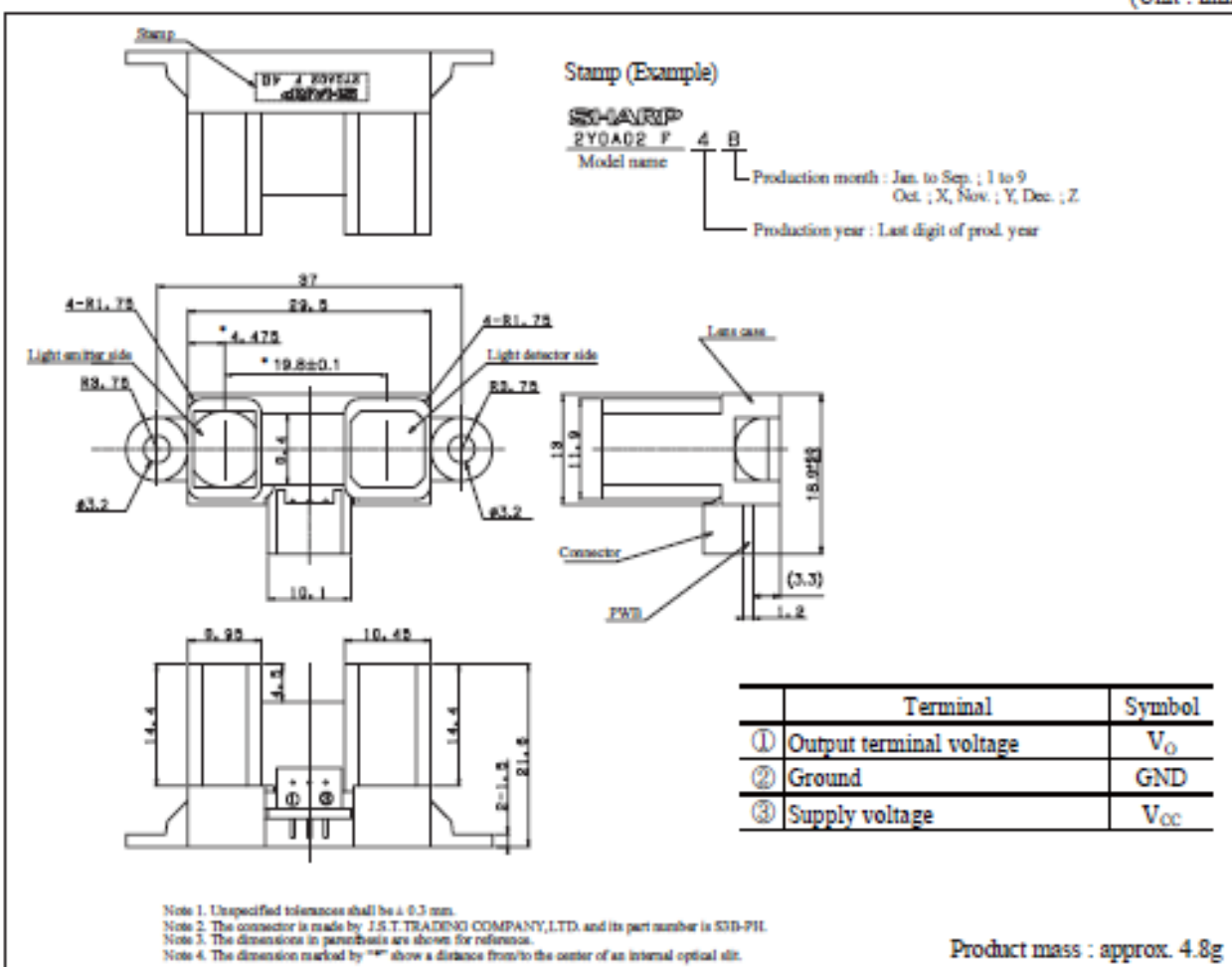
GP2Y0A02YK0F

■ Block diagram



■ Outline Dimensions

(Unit : mm)



■ **Absolute Maximum Ratings** ($T_s=25^\circ\text{C}$, $V_{CC}=5\text{V}$)

Parameter	Symbol	Rating	Unit
Supply voltage	V_{CC}	-0.3 to +7	V
Output terminal voltage	V_O	-0.3 to $V_{CC}+0.3$	V
Operating temperature	T_{opr}	-10 to +60	$^\circ\text{C}$
Storage temperature	T_{stg}	-40 to +70	$^\circ\text{C}$

■ **Electro-optical Characteristics** ($T_s=25^\circ\text{C}$, $V_{CC}=5\text{V}$)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Average supply current	I_{CC}	$L=150\text{cm}$ (Note 1)	—	33	50	mA
Measuring distance range	ΔL	(Note 1)	20	—	150	cm
Output voltage	V_O	$L=150\text{cm}$ (Note 1)	0.25	0.4	0.55	V
Output voltage differential	ΔV_O	Output voltage difference between $L=20\text{cm}$ and $L=150\text{cm}$ (Note 1)	1.8	2.05	2.3	V

* L : Distance to reflective object

Note 1 : Using reflective object : White paper (Made by Kodak Co., Ltd. gray cards R-27·white face, reflectance; 90%)

■ **Recommended operating conditions**

Parameter	Symbol	Conditions	Rating	Unit
Supply voltage	V_{CC}		4.5 to 5.5	V

Fig. 1 Timing chart

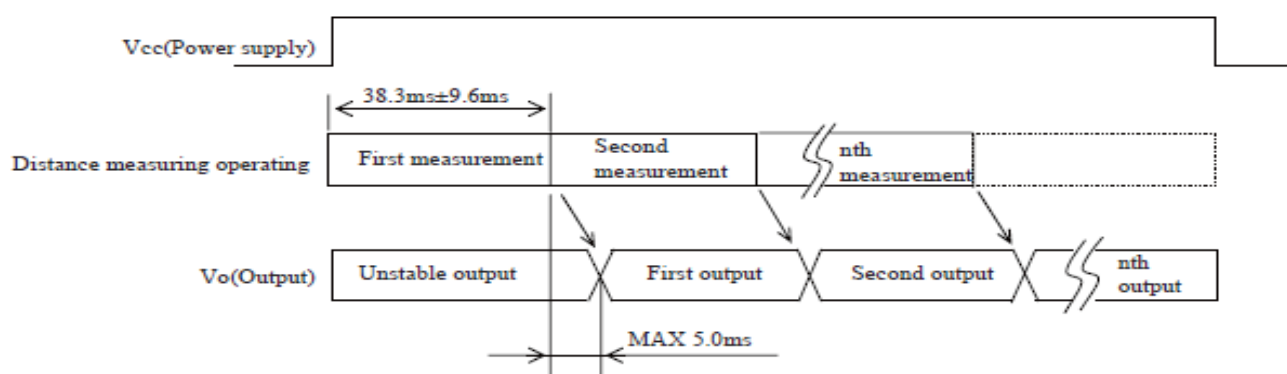
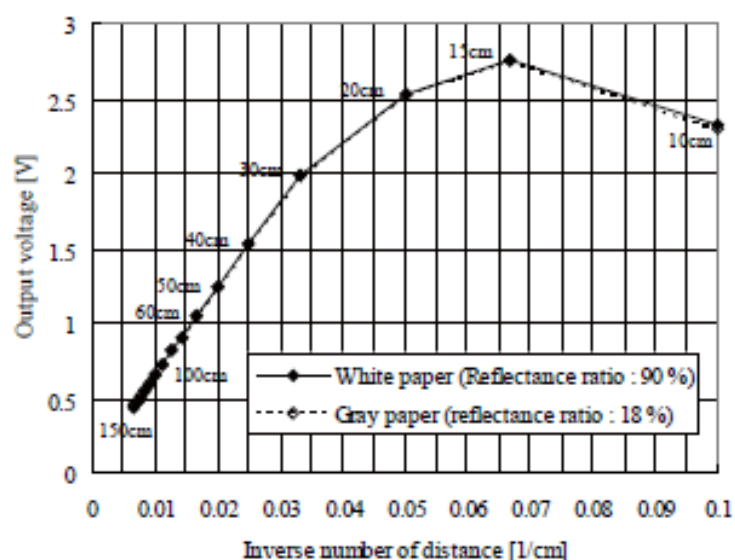
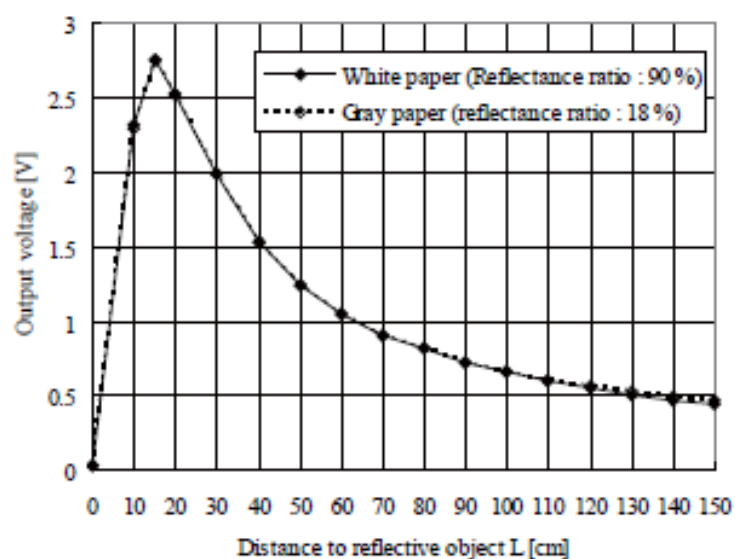


Fig. 2 Example of distance measuring characteristics (output)



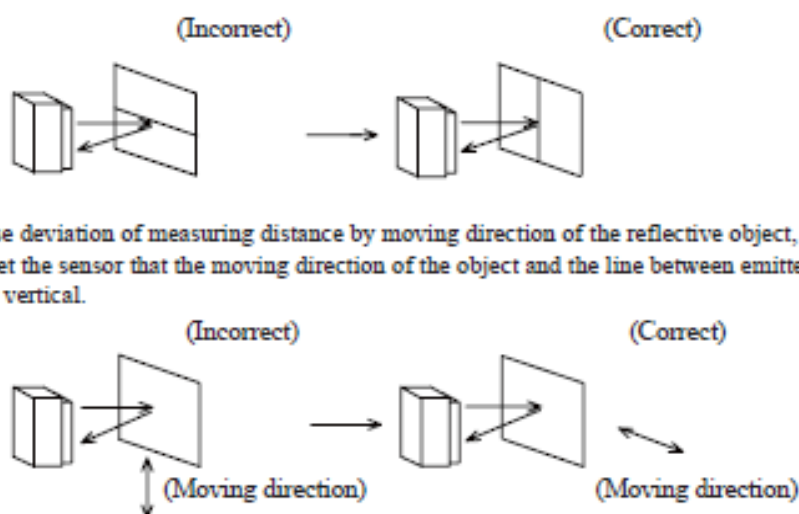
■ Notes

● Advice for the optics

- The lens of this device needs to be kept clean. There are cases that dust, water or oil and so on deteriorate the characteristics of this device. Please consider in actual application.
- Please don't do washing. Washing may deteriorate the characteristics of optical system and so on. Please confirm resistance to chemicals under the actual usage since this product has not been designed against washing.

● Advice for the characteristics

- In case that an optical filter is set in front of the emitter and detector portion, the optical filter which has the most efficient transmittance at the emitting wavelength range of LED for this product ($\lambda = 850 \pm 70\text{nm}$), shall be recommended to use. Both faces of the filter should be mirror polishing. Also, as there are cases that the characteristics may not be satisfied according to the distance between the protection cover and this product or the thickness of the protection cover, please use this product after confirming the operation sufficiently in actual application.
- In case that there is an object near to emitter side of the sensor between sensor and a detecting object, please use this device after confirming sufficiently that the characteristics of this sensor do not change by the object.
- When the detector is exposed to the direct light from the sun, tungsten lamp and so on, there are cases that it can not measure the distance exactly. Please consider the design that the detector is not exposed to the direct light from such light source.
- Distance to a mirror reflector can not be sometimes measured exactly.
In case of changing the mounting angle of this product, it may measure the distance exactly.
- In case that reflective object has boundary line which material or color etc. are excessively different, in order to decrease deviation of measuring distance, it shall be recommended to set the sensor that the direction of boundary line and the line between emitter center and detector center are in parallel.



● Advice for the power supply

- In order to stabilize power supply line, we recommend to insert a by-pass capacitor of $10\mu\text{F}$ or more between Vcc and GND near this product.

● Notes on handling

- There are some possibilities that the internal components in the sensor may be exposed to the excessive mechanical stress. Please be careful not to cause any excessive pressure on the sensor package and also on the PCB while assembling this product.

12.3 Hoja de características del driver L293

L293, L293D QUADRUPLE HALF-H DRIVERS

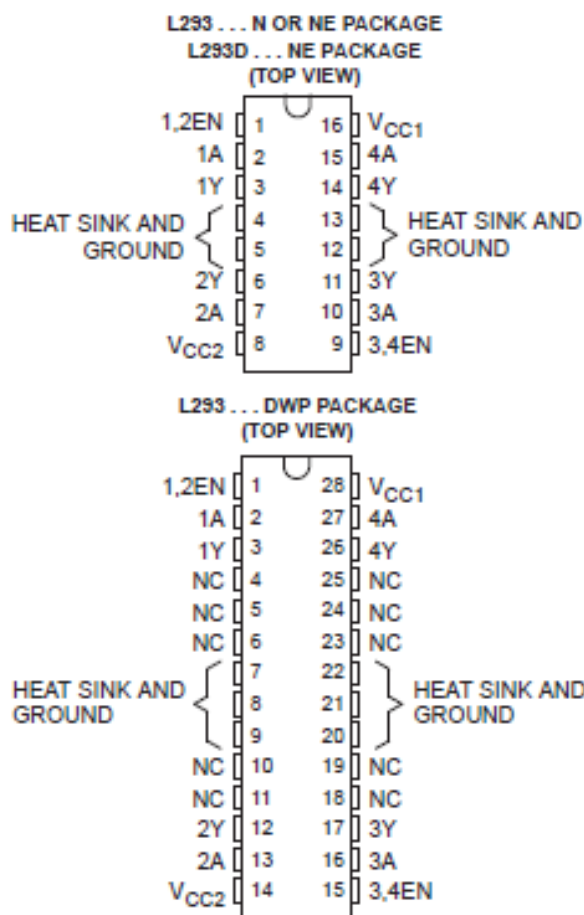
SLR3008C – SEPTEMBER 1986 – REVISED NOVEMBER 2004

- Featuring Unitrode L293 and L293D Products Now From Texas Instruments
- Wide Supply-Voltage Range: 4.5 V to 36 V
- Separate Input-Logic Supply
- Internal ESD Protection
- Thermal Shutdown
- High-Noise-Immunity Inputs
- Functionally Similar to SGS L293 and SGS L293D
- Output Current 1 A Per Channel (600 mA for L293D)
- Peak Output Current 2 A Per Channel (1.2 A for L293D)
- Output Clamp Diodes for Inductive Transient Suppression (L293D)

description/ordering information

The L293 and L293D are quadruple high-current half-H drivers. The L293 is designed to provide bidirectional drive currents of up to 1 A at voltages from 4.5 V to 36 V. The L293D is designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5 V to 36 V. Both devices are designed to drive inductive loads such as relays, solenoids, dc and bipolar stepping motors, as well as other high-current/high-voltage loads in positive-supply applications.

All inputs are TTL compatible. Each output is a complete totem-pole drive circuit, with a Darlington transistor sink and a pseudo-Darlington source. Drivers are enabled in pairs, with drivers 1 and 2 enabled by 1,2EN and drivers 3 and 4 enabled by 3,4EN. When an enable input is high, the associated drivers are enabled, and their outputs are active and in phase with their inputs. When the enable input is low, those drivers are disabled, and their outputs are off and in the high-impedance state. With the proper data inputs, each pair of drivers forms a full-H (or bridge) reversible drive suitable for solenoid or motor applications.



ORDERING INFORMATION

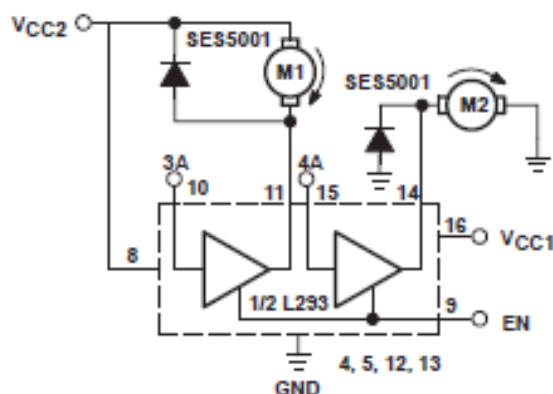
T _A	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 70°C	HSOP (DWP)	Tube of 20	L293DWP	L293DWP
	PDIP (N)	Tube of 25	L293N	L293N
	PDIP (NE)	Tube of 25	L293NE	L293NE
		Tube of 25	L293DNE	L293DNE

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.

L293, L293D QUADRUPLE HALF-H DRIVERS

SLR0008C - SEPTEMBER 1985 - REVISED NOVEMBER 2004

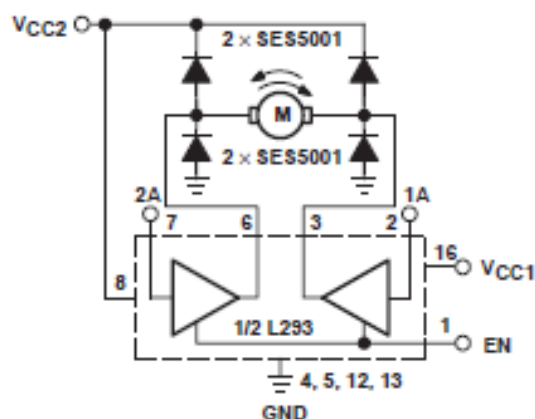
APPLICATION INFORMATION



EN	3A	M1	4A	M2
H	H	Fast motor stop	H	Run
H	L	Run	L	Fast motor stop
L	X	Free-running motor stop	X	Free-running motor stop

L = low, H = high, X = don't care

Figure 4. DC Motor Controls
(connections to ground and to
supply voltage)



EN	1A	2A	FUNCTION
H	L	H	Turn right
H	H	L	Turn left
H	L	L	Fast motor stop
H	H	H	Fast motor stop
L	X	X	Fast motor stop

L = low, H = high, X = don't care

Figure 5. Bidirectional DC Motor Control

L293, L293D QUADRUPLE HALF-H DRIVERS

SLR3008C - SEPTEMBER 1986 - REVISED NOVEMBER 2004

APPLICATION INFORMATION

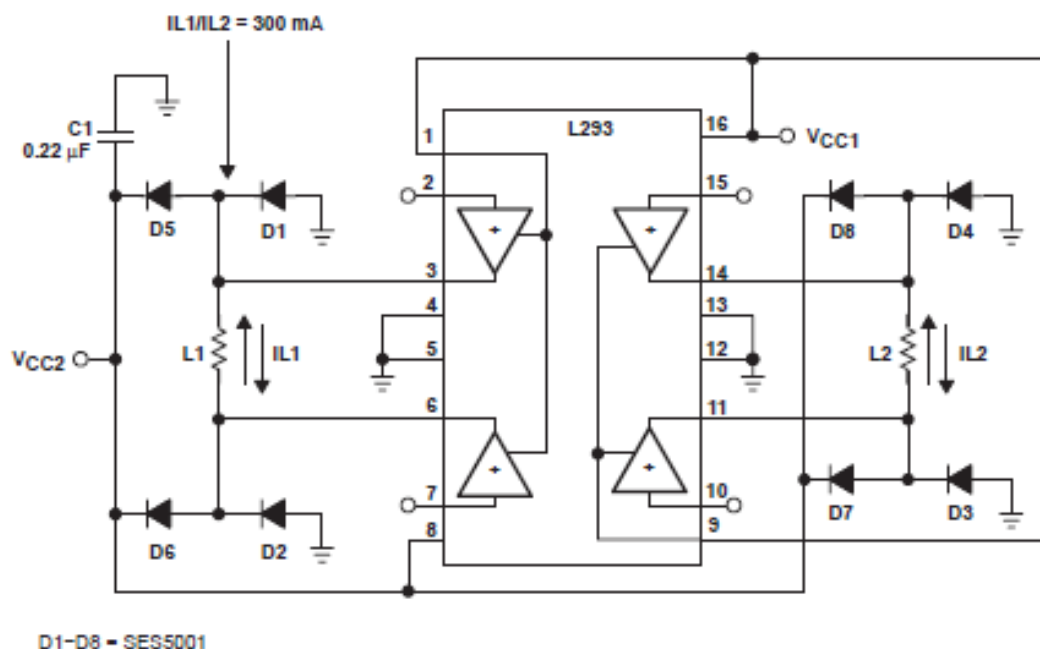


Figure 6. Bipolar Stepping-Motor Control

L293, L293D QUADRUPLE HALF-H DRIVERS

SLR3008C - SEPTEMBER 1986 - REVISED NOVEMBER 2004

APPLICATION INFORMATION

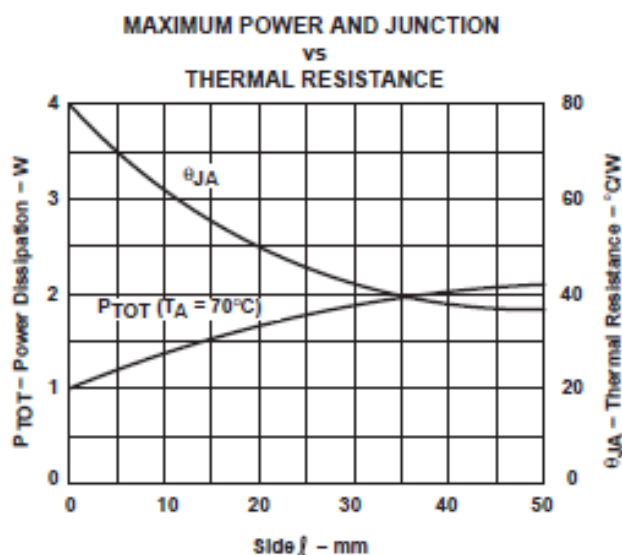


Figure 9

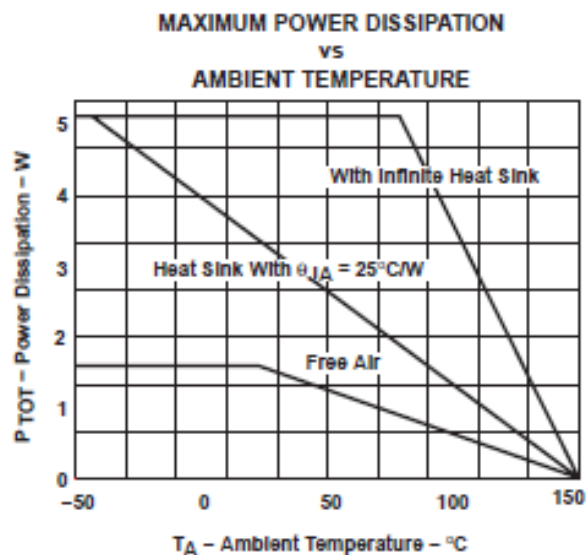


Figure 10

12.4 Esquemático de la shield

